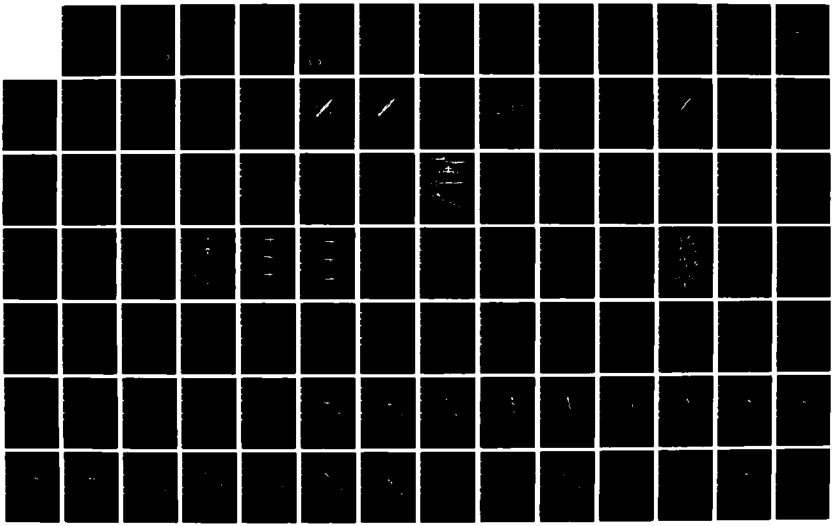
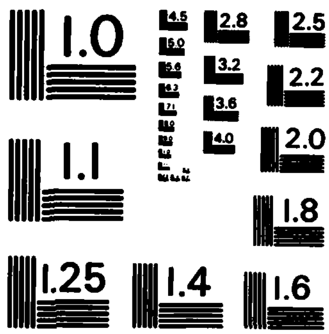
AD-A146 525 COMPUTER ANIMATED REPRESENTATIONS TO OPTICALLY OBSERVE 1/2
NUMERICAL EVALUATI..(U) AERONAUTICAL SYSTEMS DIV
WRIGHT-PATTERSON AFB OH DIRECTORATE O..
UNCLASSIFIED M J MIEDLAR ET AL. APR 83 F/G 9/2 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

(2)

ASD(ENF)-TR-83-5003

COMPUTER ANIMATED REPRESENTATIONS TO OPTICALLY OBSERVE NUMERICAL EVALUATIONS (CARTOONE)

Computer Generated Animations of Solid Bodies

MICHAEL J. MIEDLAR

RANDALL L. OLSON

Flight Technology Division
Directorate of Flight Systems Engineering

April 1983

Final Report for Period September 1981 - June 1982

Approved for public release, distribution unlimited

DEPUTY FOR ENGINEERING
AERONAUTICAL SYSTEMS DIVISION
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

NOTICE

When government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related government procurement operation, the United States Government thereby incurs no responsibility or any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

MICHAEL J. MIEDLAR	RANDALL L. OLSON	BRUCE B. KINGMAN, Chief
Aerospace Engineer	Aerospace Engineer	Flight Stability & Control Branch

FOR THE COMMANDER

ERIC E. ABELL
Technical Director
Flight Systems Engineering

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
ASD(ENF)-TR-83-5003

2. GOVT ACCESSION NO.
A146 525

3. RECIPIENT'S CATALOG NUMBER

4. TITLE (and Subtitle)
COMPUTER ANIMATED REPRESENTATIONS TO OPTICALLY OBSERVE NUMERICAL EVALUATIONS (CARTOONE) [redacted]-Computer Generated Animations of Solid Bodies

5. TYPE OF REPORT & PERIOD COVERED
Final
September 1981 - June 1982

6. PERFORMING ORG. REPORT NUMBER

7. AUTHOR(s)
Michael J. Miedlar
Randall L. Olson

8. CONTRACT OR GRANT NUMBER(s)

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Aeronautical Systems Division (ASD/ENFTC)
Wright-Patterson AFB, Ohio 45433

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Same as in Item 19

12. REPORT DATE
April 1983

13. NUMBER OF PAGES
128

14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)

15. SECURITY CLASS. (of this report)
UNCLASSIFIED

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Computer Generated Movies	Computer Graphics
MOVIE.BYU	16mm Movie
Stability and Control	MODCOMP CLASSIC
Computer Simulation	COMP80
Time History	Animated Movies

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report documents the work done in-house by personnel of ASD/ENFTC to develop a system which can generate animations of solid bodies. It contains a section which acts as a USER's Guide. The system uses MOVIE.BYU as a graphics base. Users supply a geometry model of the body of interest and a time history trajectory of the motion to be shown. The system manipulates the geometry model with the trajectory to create an illusion of three-dimensional motion. Printing successive pictures on film produces an animated movie of the motion.

FOREWORD

This report presents the system called "Computer Animated Representations To Optically Observe Numerical Evaluations" (CARTOONE) created in-house by the Flight Stability and Control Branch of the Flight Technology Division (ASD/ENFTC). CARTOONE provides a means to visually present information on dynamic systems to people unfamiliar with the system. This report contains a User's Guide and documents the work done to develop CARTOONE. The work was accomplished from September 1981 to June 1982.

A large percentage of the development work was done by two students, Mr Joel Warner and Mr Dan Sturdevant. Mr Warner, an aerospace engineering student at the University of Cincinnati, will graduate in June of 1983. He was indispensable in modifying the MOVIE.BYU software package to fit the needs of CARTOONE, and in the creation of the interface between a MODCOMP CLASSIC computer, on which we did our computations, and a COMP80 which can generate 16mm movies. Mr Sturdevant, an aerospace engineering student at Purdue University, will graduate in June of 1984. His work in support of the development work behind CARTOONE provided many additional features to the system. Without Joel and Dan, the evolution of CARTOONE would have been severely hindered.

The authors would like to thank the individuals of the Hybrid Simulation Division of the ASD Computer Center (ASD/ADS) for their help, support, and encouragement during the development of CARTOONE and during subsequent applications of our system for different tasks. We would like to thank Major Michael Wirth and the Air Force Institute of Technology for their assistance in obtaining MOVIE.BYU. We would also like to thank the people of the Electronic Printing Branch of the 2750th Air Base Wing (2750ABW/DAH) for creating our films. Furthermore, we would like to publicly thank the following people for creating MOVIE.BYU geometry models for us: David Bougine (ASD/AFEF) for a F-15 model, John Browne (ASD/YWE) for a B-1B model, Joseph Galletti (ASD/ENFTC) for a model of a Remotely Piloted Vehicle, Peggy Miedlar (ASD/ENFSS) for a KC-10 model, Richard Mutzman (ASD/ENFTC) for a model of the T-46, 2Lt Allan Netzer (ASD/TAEF) for a MIG-21 model, and Glenn Pasquini (ASD/ENFTC) for both a F-5E and a PA-48 model.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont'd)

LIST OF ILLUSTRATIONS (Cont'd)

SECTION I

INTRODUCTION

Engineers in each technical discipline use certain terms and expressions which mean very little to people unfamiliar with the discipline. For example, the value for Dutch Roll damping tells a stability and control engineer a great deal about the flying qualities of an aircraft. To a structural engineer, this value means very little. The structural engineer would be far more interested in Fracture Toughness, but this value tells very little to a survivability and vulnerability engineer, who tracks parameters such as Electromagnetic Pulse. This limits the communication between personnel in different disciplines, as well as between engineers and supervisors who are responsible for work done in more than one technical area.

In the field of Flight Stability and Control, computer simulations are used to solve many complex dynamic problems. The output of these simulations usually take the form of graphs, tables, and strip chart recordings. Someone not actively involved in the day to day operation of the simulation will have a great deal of difficulty obtaining much useful information from the tremendous volume of data that a simulation generates. An engineer trying to brief such a person on simulation results will face a serious communication problem.

This report presents a system which bridges both of these communication gaps. "Computer Animated Representations To Optically Observe Numerical Evaluations" (CARTOONE) can generate a pictorial representation describing the motion of a dynamic system. Such a representation allows the user to pass on technical information to someone unfamiliar with the problem being studied in an easily understandable way. By demonstrating the subject of interest visually, the engineer need not rely on the obscure terminology of his or her discipline, or on volumes of technical data, to transmit technical information to an interested but uninformed listener.

The user of CARTOONE must generate a time history trajectory which completely defines the motion of interest. This trajectory must contain the three translations and the three rotations of the body at successive discrete time steps. In addition, the user must supply CARTOONE with a MOVIE.BYU geometry model of the body under study. CARTOONE, using the time history trajectory, manipulates the geometry model to create an illusion of three-dimensional motion of a solid body.

This report, besides containing a User's Guide for CARTOONE, describes the modifications and additions we made to the MOVIE.BYU software. At present, CARTOONE is only available for use on MODCOMP CLASSIC computers. However, the method is easily adaptable to other operating systems. Furthermore, although CARTOONE presently can only transmit MOVIE.BYU output for interface with PLOT-10 dependent devices or in the proper format for input to a COMP80, the procedure can be easily expanded to include other devices.

SECTION II

USER'S GUIDE FOR CARTOONE

It has been said that one picture says a thousand words. CARTOONE allows the user to display 24 pictures per second on paper or on 16mm movie film. These pictures can tell volumes about the dynamics of a real world motion, passing on a wealth of information to a previously uninformed viewer. CARTOONE was written to be a user friendly computer graphics package, allowing the user to treat it like a "black box", putting in data at one end and receiving a series of pictures or an animated movie of a particular motion at the other end.

The typical user will not have an exhaustive quantity of knowledge about computer graphics. However, CARTOONE does use a MOVIE.BYU (Reference 1) geometry model of the body being animated. The model must be oriented such that the X-Y-Z axes of MOVIE.BYU correspond to the X-Y-Z motion axes of the body. MOVIE.BYU orients its axes with +X to the right, +Y up, and +Z orthogonal to this plane, out of the page. Under standard nomenclature, an aircraft +X axis is out the nose, +Y axis out the right wing, and +Z out the bottom, so the proper initial orientation of an aircraft geometry model is with the nose to the right, and right wing pointed up. The proper arrangement is illustrated by a geometry model of an F-4 Phantom in Figure 1. This figure shows the orientation of the axes, and shows the bottom of the aircraft. The model should be positioned so that the center of rotation of the body (usually the center of gravity) coincides with the origin of the geometry model. CARTOONE assumes that the model is properly oriented, and manipulates it accordingly. The user has the responsibility of ensuring that the model is oriented properly before using CARTOONE.

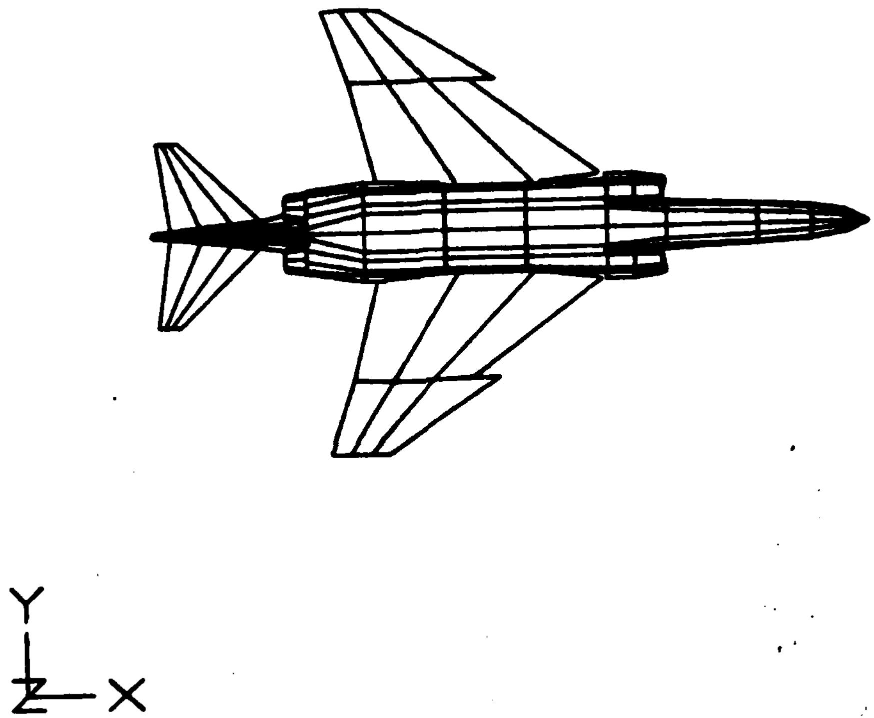


FIGURE 1
INITIAL ORIENTATION OF AXES FOR CARTOONE

The user must also supply a description of the motion to be shown, in the form of a time history trajectory file. This file should contain the three translations and the three rotations of the aircraft at fixed time intervals. CARTOONE reads the parameters in the following order:

1. Time.
2. Translation in the -Z direction (altitude).
3. Rotation about the +X axis (roll).
4. Rotation about the +Y axis (pitch).
5. Rotation about the +Z axis (yaw).
6. Translation in the +X direction (forward motion).
7. Translation in the +Y direction (side motion).

CARTOONE uses a free format READ statement for trajectory input, and draws a picture for each interval of the trajectory file. Depending on the complexity of the geometry model, MOVIE.BYU can generate a picture within 40 clock seconds. If the user wants the pictures to be drawn on a TEKTRONIX 4014 screen, the interval between time history data points should be large enough to complete the animation in a reasonable length of time. Sitting at a terminal watching the computer generate line drawings gets very tiring, and a one-half second interval between data points will probably be sufficient for a decent animation without overly trying the user's patience. If the object is to create a movie, the trajectory time interval should be 1/24 second per frame, the inverse of the 24 frames per second run speed of 16mm movies. In all cases, the time steps between trajectory intervals should be constant to ensure a smooth animation.

The CARTOONE software package uses several I/O units to perform its input/output functions. Prior to invoking CARTOONE, the user must connect the units with the proper local files. These units, their functions, and the appropriate files are:

Unit No.	Function	File
5	User Inputs	Terminal
6	MOVIE.BYU Output	Terminal
7	Geometry File Input	Local File Containing Geometry Model
13	Trajectory File Input	Local File Containing Trajectory File
14	COMP80 Output	Magnetic Tape
16	Scratch File	Any unused Local File

When running CARTOONE at a terminal, all units except 14 are used. When writing output for a COMP80, unit 14 must be assigned to the tape unit.

CARTOONE has a series of run time options available to the user. After creating the geometry and trajectory files, the user invokes CARTOONE and answers the questions it asks pertaining to the options available. If CARTOONE is drawing the pictures on a TEKTRONIX 4014 screen, the user must perform these tasks by hand. When creating a movie, the user will find it more convenient, as well as more efficient, to run CARTOONE through the computer batch system. To do this, the user will need to create a local file containing the proper statements to invoke CARTOONE and choose the required run time options. The questions CARTOONE asks and the available options are listed in Table 1 at the end of this section. The user will need to create a procedure file to link CARTOONE with this command file. CARTOONE reads command inputs through Unit 5, and the user will need to set this unit to the command file, rather than the terminal input file. The user should also set Unit 6 to an unused local file for COMP80 animations.

For convenience of operation, CARTOONE functions as an enhancement of the DISPLAY software of MOVIE.BYU. This enhanced form does all the things that standard MOVIE.BYU does, and contains all the components of CARTOONE as well. To the uninformed, the two versions will operate identically, since only by entering the proper commands do the differences appear. For purposes of this report, "MOVIE.BYU" will represent the standard DISPLAY software of the universal system, and CARTOONE will refer to the system which manipulates DISPLAY to generate animations. The two software packages do not act independently in our system, but do appear to function as two separate systems and the user can treat them as such. The user can write a procedure file to ensure that the proper unit assignments are made, and the same procedure file can then invoke DISPLAY in the normal way.

MOVIE.BYU operates normally until the user enters "CART", the command which invokes CARTOONE. CART is an additional allowable command of our enhanced version of MOVIE.BYU. CARTOONE responds by asking the user to enter the required output device, and will accept as proper responses either 4014, 4663, or COMP. If the user specifies 4014, CARTOONE sends the MOVIE.BYU output through Unit 6 in the proper manner of writing to a TEKTRONIX 4014 terminal. Entering 4663 causes the output to be written to a Tektronix 4663 flatbed plotter. Option COMP causes CARTOONE to write the output through Unit 14 in the proper formats for writing to a magnetic tape as input to a COMP80 device. Any other response to this question causes it to be repeated.

CARTOONE next asks which of the six viewing modes it should use, and only accepts F (Fixed Position), C (Chase Plane), I (Independent Vehicle), W

(Wingman), P (Pilot Eye), or M (Mural Poster Plots), in reply. Any other answer causes CARTOONE to repeat the question. With the first option, Fixed Position, the observer position does not change while he watches the body move. The user may choose Chase Plane, in which the observer moves parallel to the original velocity vector of the body and observes the motion of the body. The first two blocks of data in the trajectory file determine the vector of motion. CARTOONE calculates the changes in X, Y, and H (altitude) and divides each by the time interval between the two blocks of data to define "velocity components" in each direction, then rewinds the trajectory file. For each frame, it multiplies the X, Y, and Z "velocities" by time to obtain the current position in the X, Y, and Z directions of the "chase plane", and moves the viewing position to this spot. The Independent Vehicle mode allows the observer to move along a trajectory which does not depend on the test aircraft. The trajectory file must contain data on both the aircraft and observer for this mode to be used. The first seven parameters should be the first interval for the observer, the second seven should be the first interval for the aircraft, and so on. The observer does not roll even though data for this rotation must be supplied. Wingman keeps the viewing position a constant relative distance from the origin of the body. For each of these viewing modes, the view remains centered on the body. Pilot Eye, created for use with aerodynamic studies of piloted vehicles, puts the viewing position at the origin of the moving body. The body itself is not seen, but the terrain moves and rotates to create the illusion of a view from the cockpit of a moving aircraft. Mural generates a Poster Plot of the motion. This can only be used with output device 4014 or 4663. Mural causes CARTOONE to overlay the pictures for each time step on one page. Experience has shown that a one second interval between pictures is usually large enough to prevent pictures from overlapping but small enough to adequately describe the motion. The direction of view is held constant. Poster Plots are dealt with in more depth in Section IV, where an example is shown. Appendix A shows a sample illustration from each animation viewing mode.

After choosing a view mode, the user is asked to supply the part limits of the MOVIE.BYU model which define the terrain. The proper reply is a pair of numbers, a lower and upper limit, such as 1,1. CARTOONE, originally created for use with aircraft six-degree-of-freedom simulations, uses a terrain model to create the illusion of three-dimensional motion by depicting the aircraft moving past a fixed reference. The terrain can be as simple as a grid of squares, and should be one part of the MOVIE.BYU geometry model with the center of the ground at the origin of the model. The horizon should be in the X-Z plane. If no terrain is required, the user should enter a pair of numbers which do not define any part of the model (i.e., if the model has 10 parts and no terrain, enter the terrain limits as 11,11 or 12,12 or some other such combination).

The user will then be asked for the part limits of the first body. CARTOONE can manipulate up to 19 independent bodies, but the picture will always be centered on the first body. The user will be asked for part limits of successive bodies until the user returns a pair of zeros. This option, Multiple Independent Vehicles, can be used to show an aircraft firing a missile, such as the F-5E before and after firing a sidewinder missile in Figures 2 and 3. Each independent body must have a trajectory describing its motion. The center of rotation of each body must originally coincide with the origin of the entire model, and must be in the proper original orientation for CARTOONE as described earlier in this section. The various trajectory files

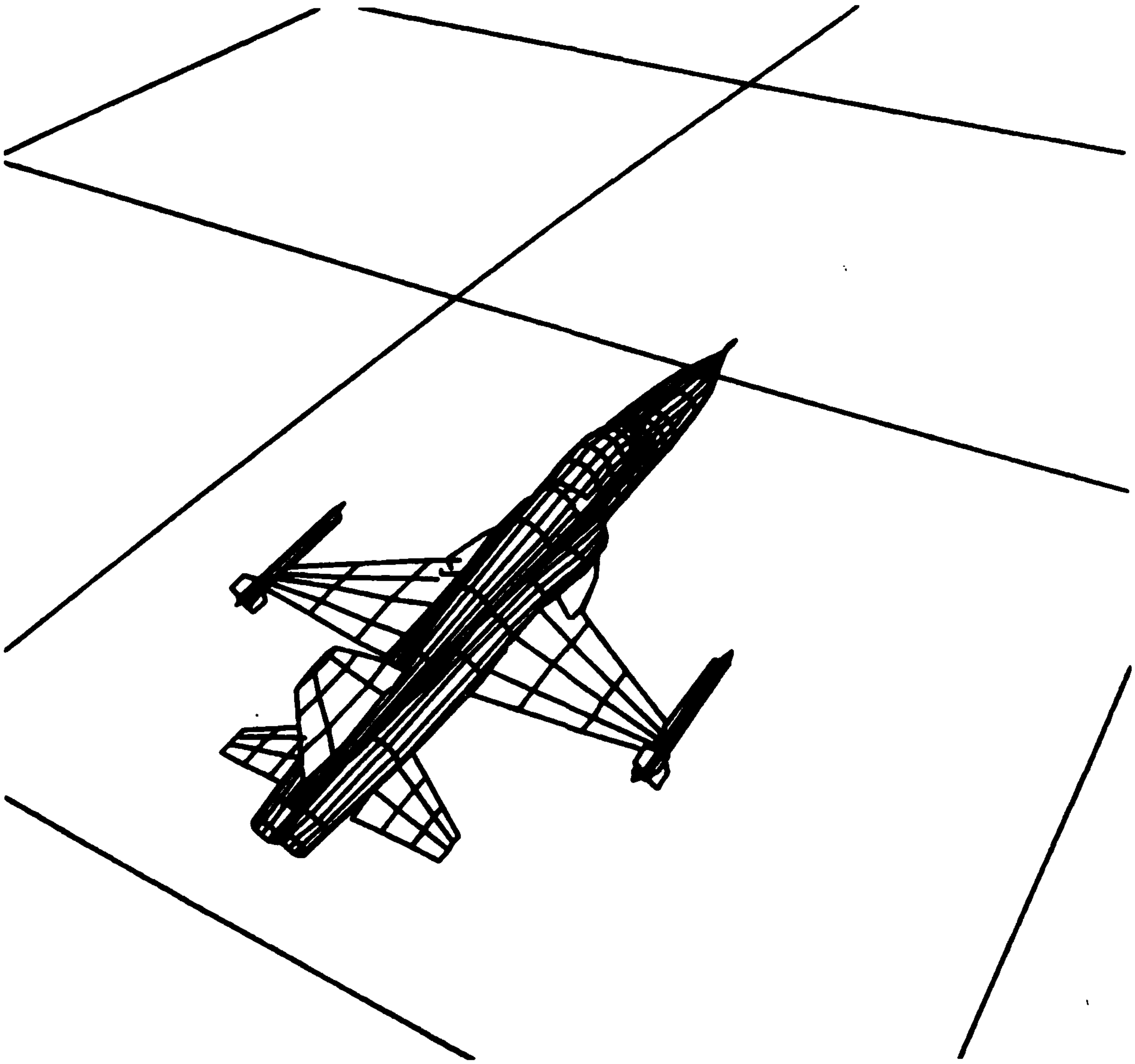


FIGURE 2
AIRCRAFT WITH MISSILE ON WINGTIP

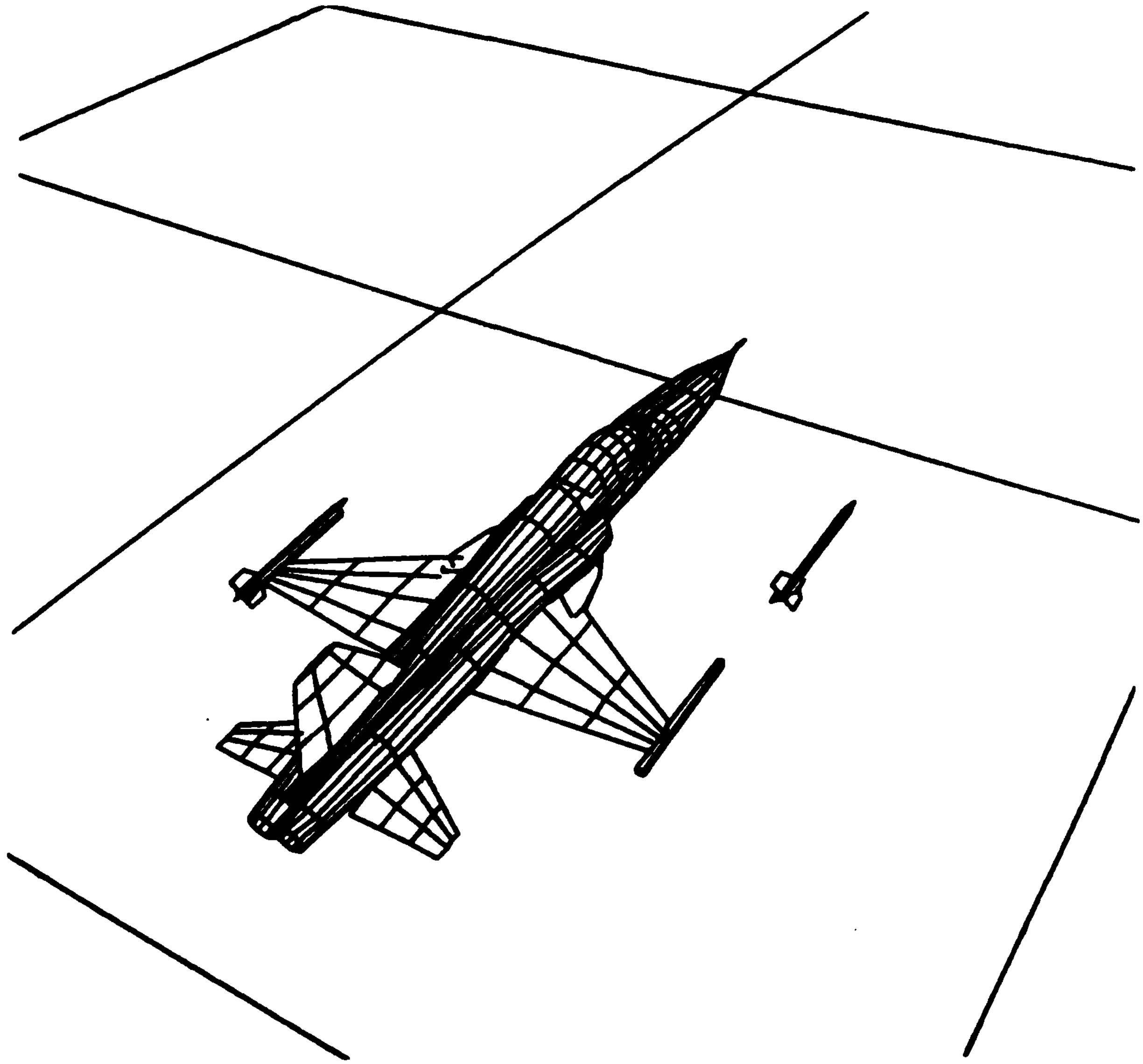


FIGURE 3
AIRCRAFT AFTER FIRING MISSILE

must be blended into a master trajectory file in sequential order. The first seven parameters of the file should be the first seven time history terms of body one, the next seven should be the first seven terms of body two, and so on. When the first time interval has been read for each body, and the first picture drawn, the same parameters will be read in the same order for the second time step, and so on through the trajectory file. A simple FORTRAN program will blend separate trajectory files into a master file. Simply set Unit 1 to the first trajectory file, Unit 2 to the second, and so forth, then read the first seven parameters from each with a free format READ statement. Use a free format WRITE statement to write the parameters to an unused unit. Do this until an end of file mark is encountered on one of the trajectory files. Write an end of file on the master trajectory and catalog it into permanent memory storage.

After reading the part limits for each body, CARTOONE will ask the user if he or she wants to use a control block. Answering "Yes" to this question invokes the Control Block option, which graphically presents the pilot control inputs on the lower left corner of the screen, as shown by the F-16 Fighting Falcon in Figure 4. This option was developed for use with aircraft motion studies. The cross presents pilot longitudinal and lateral control stick (or wheel, or column) motions as fractions of the total available, and the bar below the cross presents rudder pedal deflection as a fraction of total available. The control block presents the control inputs for the primary (first) vehicle. The trajectory file should contain lateral stick, longitudinal stick, and rudder pedal positions as the eighth, ninth, and tenth parameters for each time step of the primary trajectory. These should be fractions of maximum deflection available. For example, if the

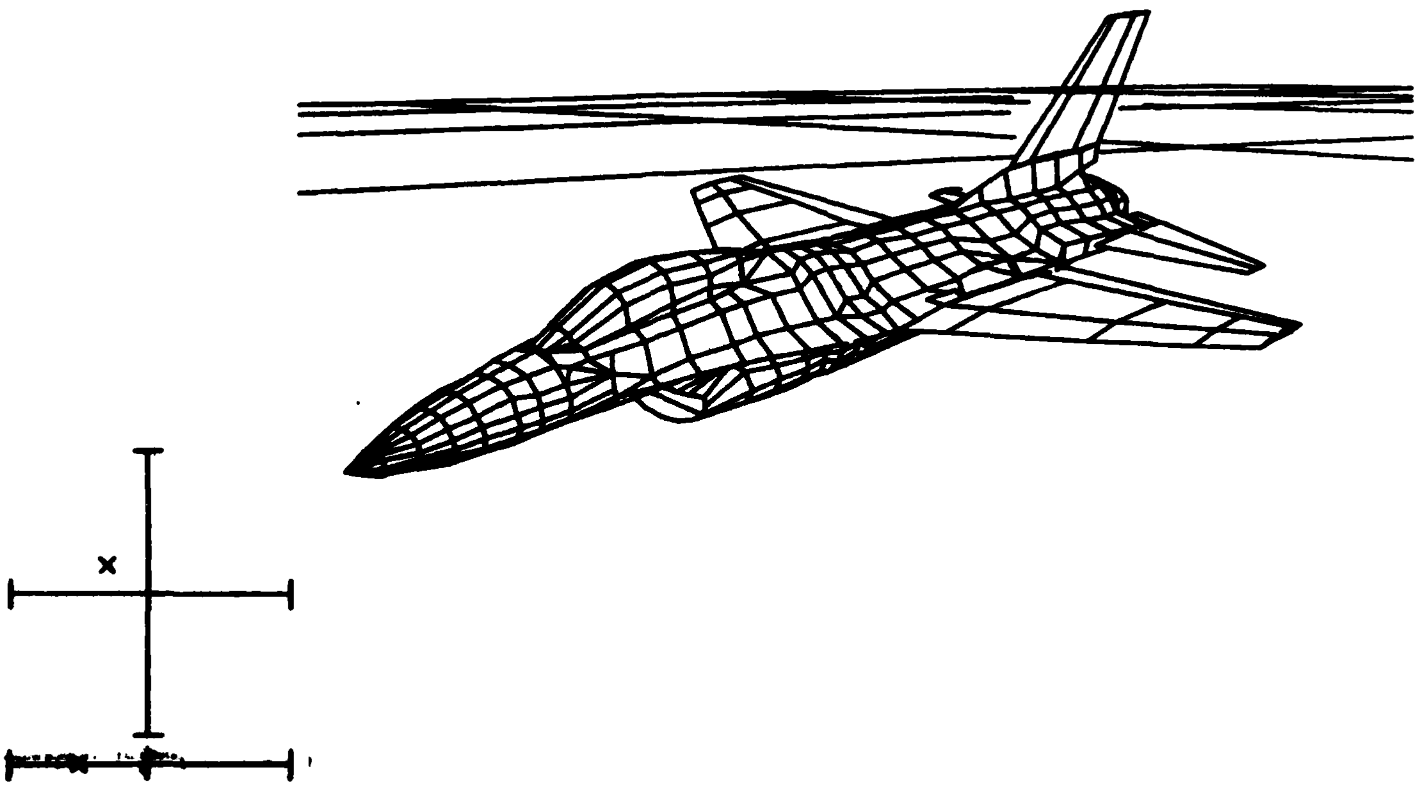


FIGURE 4
GEOMETRY MODEL WITH CONTROL BLOCK

longitudinal stick can move aft five inches, and the pilot moves it three inches aft, the value passed to CARTOONE should be 0.6. Lateral stick is positive to the right, longitudinal stick positive aft, and rudder pedals positive to the right.

After inquiring about the need for control output, CARTOONE asks for the Zoom Factor, a value which CARTOONE divides into the distance between the viewer and the primary vehicle. The nominal value is one. If greater than one, the distance reduces and the model looks closer. If less than one, the distance increases and the model appears farther away. Entering a value less than or equal to zero will cause CARTOONE to repeat the request for a Zoom Factor. Figures 5 and 6 show a B-1B Strategic Bomber with Zoom Factors of one and 15.

For View Modes Mural or Fixed Position, the user will next be asked for the initial position of the observer, the observer location at time T = 0. For viewing modes Chase Plane, Wingman, or Independent Vehicle, the user will next be asked for the initial offset position of the observer. This defines the position of the observer relative to the aircraft at time T = 0. For any of these cases, subsequent positions of the observer will depend on the view mode chosen. Choosing the Pilot Eye mode automatically places the viewer at the center of the body, so the observer position moves with the body.

CARTOONE presents the final option only if the user chooses output device COMP. The user will be asked if he wants to specify one or more Title Pages to precede the animation. These can contain anything from the title of the film to an in-depth description of the motion being displayed. CARTOONE generates each Title Page for 144 consecutive frames, which represents six seconds worth of 16mm film, and each consists of four lines of 20

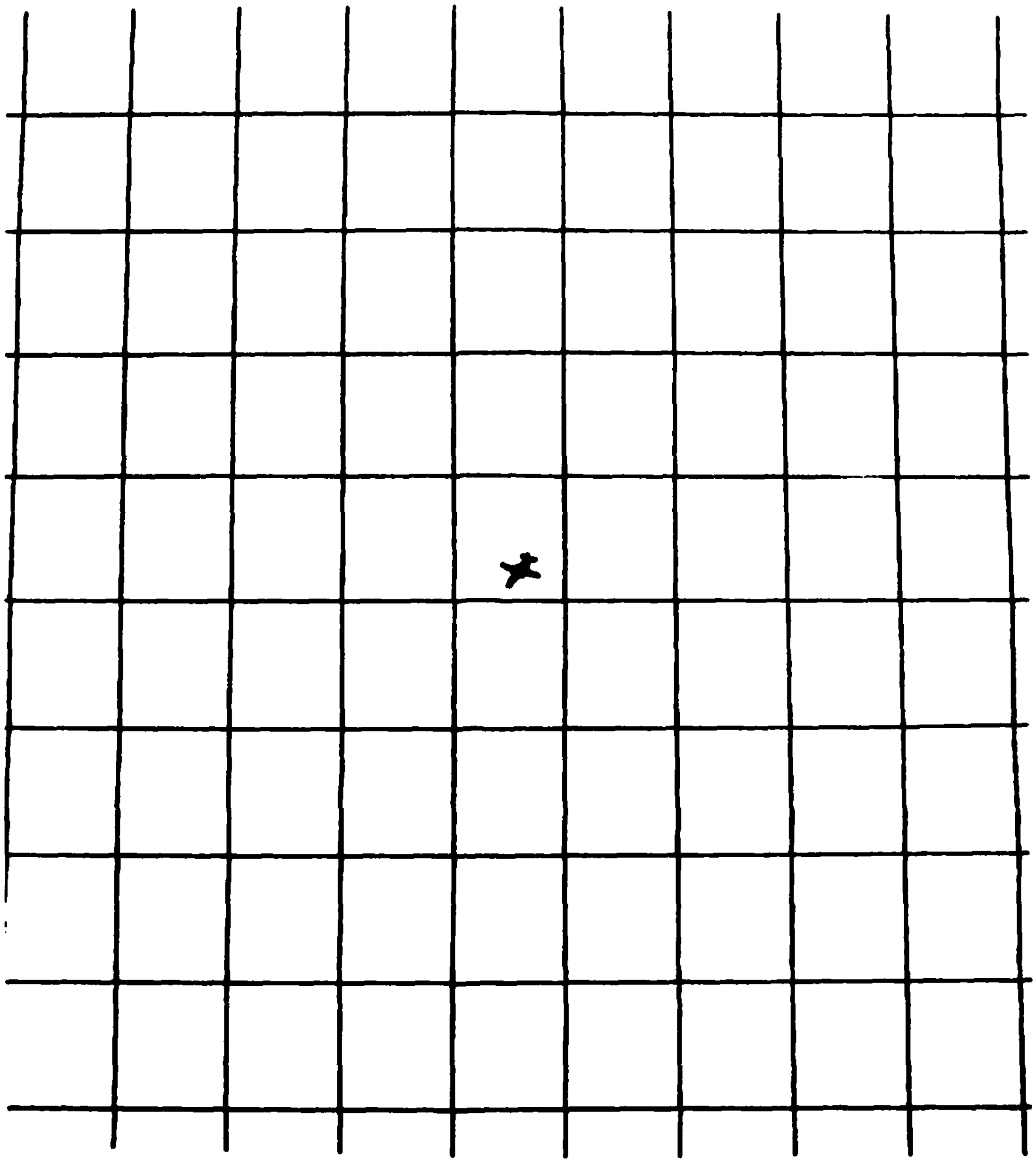


FIGURE 5
AIRCRAFT 5000 FEET AWAY - ZOOM FACTOR 1

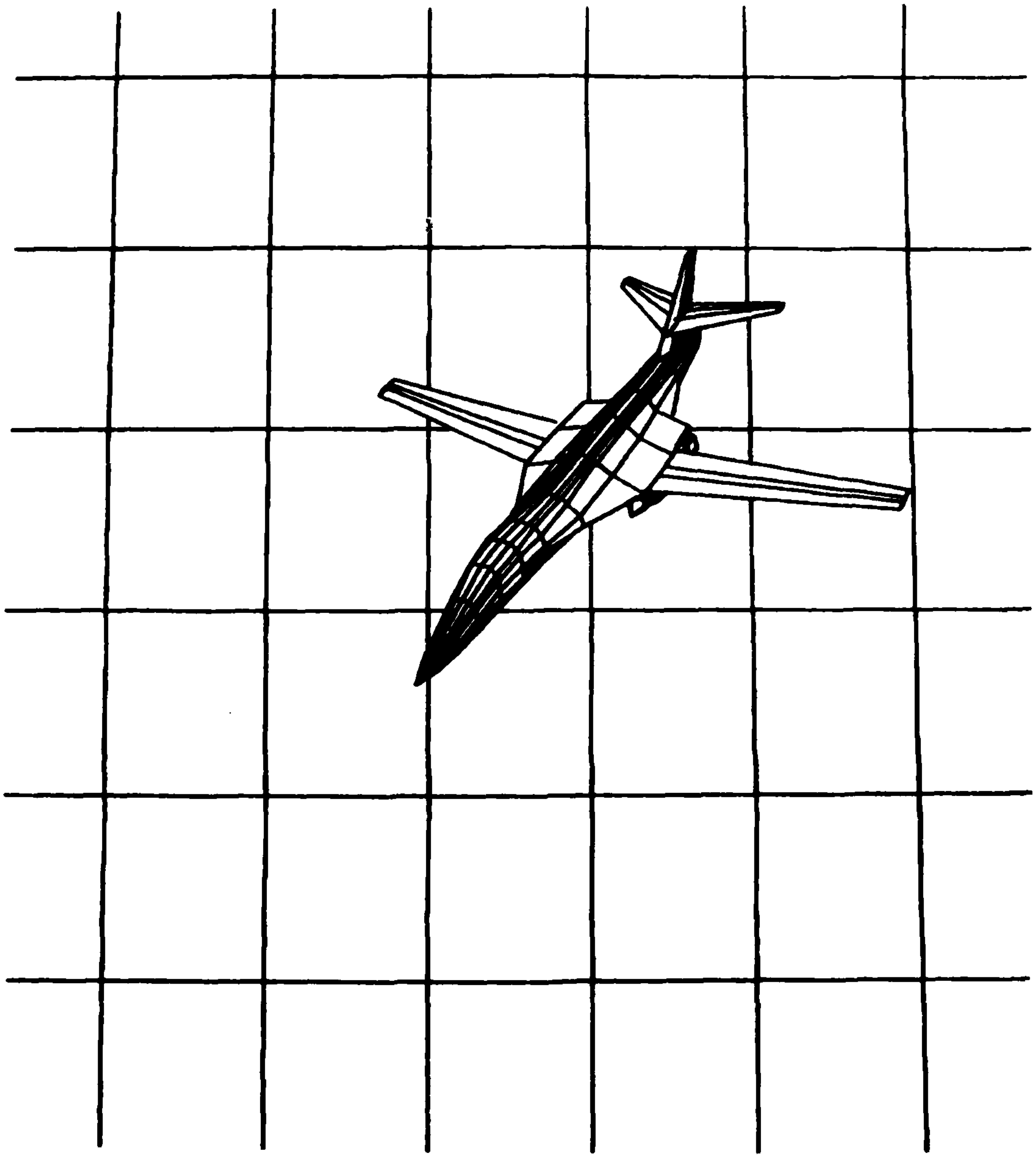


FIGURE 6
AIRCRAFT 5000 FEET AWAY - ZOOM FACTOR 15

characters each. The viewer can use the Title Pages feature to pass information directly to the viewer, rather than through the animation. The characters available for Title Pages are the capital letters of the alphabet, the numbers 0-9, a period, a comma, a dash, and an apostrophe.

One feature available to the user but not an implicit part of CARTOONE is to show motions faster or slower than real-time. Some especially rapidly developing motions cannot be adequately observed at a real-time speed, and the animation must be shown in slow motion to be properly presented. Some motions develop very slowly, and showing the motion faster than real-time might be of use. To do this, the user should construct the time history trajectory file with a time interval greater or less than 1/24 of a second. A 16mm movie projector always shows 24 frames per second of elapsed clock time. CARTOONE draws one frame for each step in the trajectory. To create a film showing the motion at one-half speed, the time interval should be 1/48 of a second, causing CARTOONE to generate twice as many frames for each real-time interval. The projector runs at the same speed, showing the motion over a longer length of time. For a film at twice real-time, the time interval should be 1/12 of a second.

The run time options must be specified prior to each animation, that is, for each independent execution of a master trajectory file. The same trajectory file can be used several times with different combinations of options to show the same motion from several view modes or angles.

If the user intends to specify output device COMP, he has several additional tasks to perform. The output unit of the COMP80 interface, 14, must be connected to a tape drive which writes 1600 bits per inch on a nine track magnetic tape before invoking MOVIE.BYU. When CARTOONE finishes, the user must write two end of file marks on the tape prior to rewinding it. By generating a series of tapes, the user can create a movie of any length. The tapes carry the MOVIE.BYU output to a COMP80 output device, which generates the film. This device can also be treated as a black box; the user inputs tapes and gets back a 16mm film.

For viewing at a Tektronix 4014 or 4663, the user must direct the CARTOONE output unit, 6, to the terminal local output file before invoking MOVIE.BYU. The user inputs his choice of run time options. CARTOONE draws the first picture, then pauses until the user returns a blank. This pause gives the user a chance to generate a hard copy of the picture on the screen. This sequence of pictures and pauses continues through the entire trajectory file. Returning any character other than a blank causes the animation to cease and normal MOVIE.BYU activity to resume.

CARTOONE has another feature for the user who works at a Tektronix 4014 terminal called "COMBINATION" or "COMB." The COMB subroutine contains a series, or combination, of commands which make the task of creating animations on paper easier. The COMB features are listed in Table 2 at the end of this section. After invoking CART, the user returns to MOVIE.BYU by entering any character other than a blank during a pause following the drawing of a picture. COMB is another allowable command of our special form of MOVIE.BYU. After invoking COMB, the user can, by entering a blank, return to CART and resume the animation at the point where the user returned to MOVIE.BYU. Other possible commands under COMB include RETURN, REWIND, SKIP, PAUSE, DATA, HELP, ?, and DEBUG. RETURN returns the user to MOVIE.BYU. REWIND rewinds the trajectory file to the first time step. SKIP allows user to jump ahead through the trajectory to a time step of interest. If the user specifies SKIP, COMB asks "UNTIL WHAT TIME?", then waits for the user to enter the time step of interest. COMB moves forward through the trajectory file to the first time interval greater than or equal to the time specified, returns to CART, and pauses. Entering a blank restarts the animation at that point. PAUSE returns the user to CART, but does not draw the next picture until the user returns a blank. The DATA feature displays the current values of the time history trajectory parameters for the primary vehicle and the current observer position. HELP and ? both list all possible options of COMB. DEBUG gives the user the option of entering trajectory data manually, causing CARTOONE to read data (i.e., trajectory information) from the terminal screen instead of a trajectory file. The user can utilize DEBUG to view a particular motion without actually building a complete time history trajectory file. Invoking DEBUG once changes the trajectory input unit to the terminal input file, and invoking it again changes the trajectory input unit back to its default area.

The average user of CARTOONE may not have, and will not require, much prior knowledge of computer graphics. If a geometry model of the body of interest already exists, the user does not really need to know anything at all about computer graphics. However, a working knowledge of the general functions and operations of MOVIE.BYU would be an asset. As the user becomes familiar with CARTOONE, he will develop his own uses for the system. It is tremendously versatile, and can be used for a large variety of applications, easing the task of engineers in all disciplines, especially when communicating with management.

TABLE 1

SEQUENTIAL ORDER OF CARTOONE FEATURES

FEATURE	OPTIONS
Output Device	4014, 4663, COMP 80
View Mode	Fixed, Chase, Wingman, Independent, Pilot, Mural
Terrain Part Limits	Lower Limit, Upper Limit
Vehicle Part Limits	Lower Limit, Upper Limit or 0,0
Control Block	Yes, No
Zoom Factor	Any Number Greater Than Zero (Nominal = 1)
Initial Orientation (for M or F)	X, Y, H
Initial Observer Offset (for C, W, or I)	DX, DY, DH
Title Pages (for COMP80)	Yes, No

TABLE 2

COMB FEATURES

FEATURE	FUNCTION
blank	Re-enter CART, draw next frame
DATA	Print current trajectory and observer information
DEBUG	Switches trajectory source between terminal and file
HELP or ?	Lists COMB features
PAUSE	Re-enter CART, pause until blank is returned
RETURN	Return to MOVIE.BYU
REWIND	Rewind trajectory file
SKIP	Skips ahead to specified time step

SECTION III

MOVIE ON THE MODCOMP

CARTOONE operates on a MODCOMP CLASSIC (hereafter simply referred to as a MODCOMP) computer, a super-mini class of computer. It uses a 16 bit word with 32/64/128 bit hardware multiply. Two, four, or eight words can be used to create a floating point number, and one or two words can be used to make integers. A standard block of MODCOMP memory consists of 64K 16 bit words, where K equals 1024. Typical programs can access two such blocks. During execution, data fills one block of memory, and pure (or re-entrant) code fills the other. Pure code does not change during execution and can be shared between users, if the code is globally available.

Following the recommendations of Major Michael Wirth, the Aeronautical Systems Division (ASD) Computer Center purchased Version 3 of MOVIE.BYU from Dr. Henry Christiansen of the Brigham Young University. This version, written for use on a DEC-10, can draw a model with a maximum of 250 nodes (or joints) and 250 polygons (or elements). Brigham Young University has sold this version of the software to over 700 organizations in 22 countries. The complete software package consists of seven individual files, called DISPLAY, UTILITY, MOVIE.DOC, SECTION, MOSAIC, TITLE, and UPDATE. The system we modified used only three of these files, DISPLAY, UTILITY, and MOVIE.DOC. DISPLAY generates the images, and when we refer to "MOVIE.BYU" we actually mean DISPLAY. UTILITY creates and manipulates geometry files, and its final product is a file which DISPLAY can draw. MOVIE.DOC (Reference 1) contains all of the documentation for the MOVIE.BYU software system. After completing testing on a DEC-10 owned and operated by the Air Force Wright Aeronautical Laboratories (AFWAL), DISPLAY, UTILITY, and MOVIE.DOC were transferred to a Control Data Corporation Cyber 74 computer owned and operated by the ASD Computer Center. The software used DEC dependent FORTRAN read statements, which had to be replaced by standard FORTRAN read statements. To run on the Cyber 74, the system needed 130K octal memory, an amount much larger than most users could access during interactive operation. After segmentation, it would run with only 65K octal memory, an amount available to the average user at Wright-Patterson Air Force Base. The next step took the Cyber 74 version of DISPLAY and UTILITY software to the ASD Hybrid Simulation Division MODCOMP computers, where the source code for DISPLAY was divided into four parts, called MAIN, MOVIEA, MOVIEB, and MOVIEC. This division provided ease of handling in the system source editor. MAIN contained the primary control program, and the subroutines to perform Euler rotations. MOVIEA and MOVIEB held the subroutines which perform global rotations, scaling, clipping, transformations, and the Watkins hidden line algorithm. MOVIEC contained device dependent drivers such as a TEKTRONIX vector terminal interface, a flat bed plotter interface, and run length encoded output file. This file provided the capability of writing to a tape for off line color device operation which use eight bits for each pixel of each of three primary colors. This system was placed onto the MODCOMP shared disk as a global file for use by any and all interested users. This version allows a maximum of 980 nodes (or joints).

For various Stability and Control applications, we use more complex models than this global MODCOMP system allows. To increase the node limit, we use MODCOMP Extended Memory to enlarge the amount of addressable memory available. Each block of Extended Memory provides an additional 64K words of memory space, and the first block increased the maximum number of nodes to 1500. Each additional block provides the capability for approximately 1000 additional nodes. Using Extended Memory increases run time somewhat, but this increase is small enough to pass unnoticed by the user. To move this software to another system the statements which put common blocks into Extended Memory have to be removed, and the addressable memory increased by some other machine dependent method.

Stability and Control work necessitates the use of Euler angle rotations of an aircraft to determine its motion as a function of time. Airplane Flight Dynamics and Automatic Flight Controls (Reference 2) contains an in-depth discussion of Euler angle rotations, and they will not be treated here. Suffice it to say that MOVIE.BYU cannot perform such rotations because the system only rotates geometry models about the three primary axes, which remain fixed, and does not have the capability of rotation about an arbitrary axis. To provide such a capability, the MODCOMP version of MOVIE.BYU has an Euler rotation option. When invoking the MOVIE.BYU ROTATE command, the user returns "E" instead of one of the three primary rotational axes. The user next enters a set of three numbers which define successive rotations about the model body axes, which translate and rotate with the model. The number one indicates a rotation about the body X-axis, two a rotation about the body Y-axis, and three a rotation about the body Z-axis. The sequence three, two, one indicates rotations about the Z axis, then a Y rotation, and finally about X (the classical yaw-pitch-roll sequence of Stability and Control). The rotations can be any sequence of Euler-like rotations (e.g. yaw-pitch-yaw, roll-pitch-yaw, etc.). MOVIE.BYU then reads the magnitudes of each rotation. If fewer than three Euler rotations are desired, dummy rotations of magnitude zero will provide the proper motion.

Because it uses Extended Memory, this version could not be made globally available, but it was available to any interested user on a personal basis.

This version of MOVIE.BYU attracted a great deal of interest. It allowed more nodes than the more publicized version on the Cyber 74, the computer used by most engineers in the Flight Technology Division. The turn-around time for problem solving was far less with the MODCOMP system than the Cyber 74 system. Many people submitted suggestions for additions to and applications of the modified software. Some people wanted static pictures of one or more aircraft to show physical relationships between aircraft flying in close proximity to each other. The MODCOMP version of MOVIE.BYU lets the user show several aircraft in arbitrary relative positions from any angle or distance, and these pictures could be studied to analyze the aerodynamic effects of the aircraft on each other. For example, the pictures in Figure 7 were generated during a study to determine the aerodynamic effects of the flow induced by the KC-10 Extender on the E-4 Command Post during aerial refueling. The version of MOVIE.BYU on the MODCOMP using one block of Extended Memory provided the capability to do this kind of work.

However, many more people wanted a dynamic system which would provide a way to demonstrate the motions of aircraft with time. No real world system is static, and static representations do not give a good idea of the dynamic motions of the system. Thus, we developed the system we call CARTOONE.

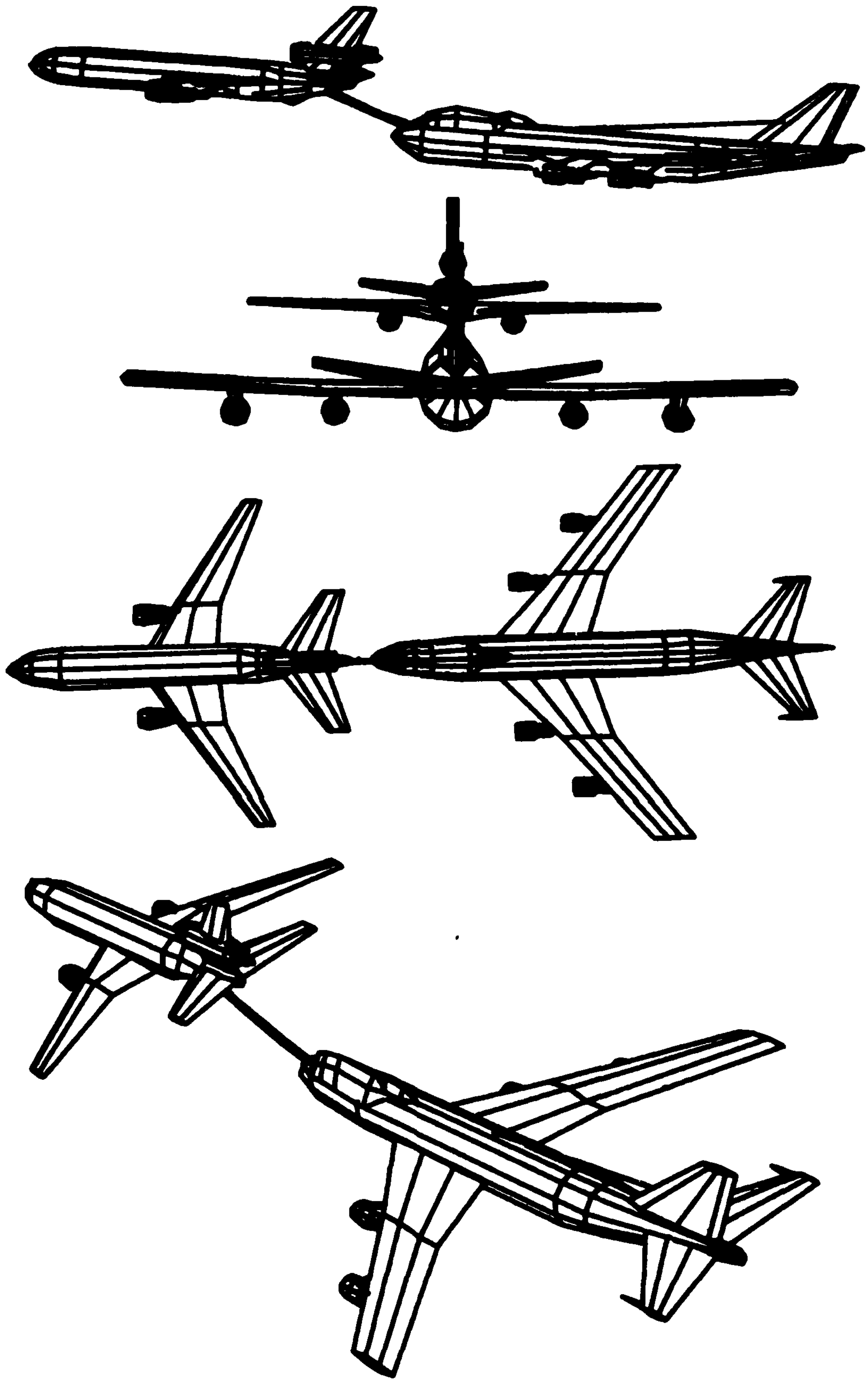


FIGURE 7
PICTURES FOR STUDY ON AERODYNAMIC
INTERACTION BETWEEN AIRCRAFT

SECTION IV

THE EVOLUTION OF CARTOONE

MOVIE.BYU generates scaled drawings of a solid body with three dimensional perspective. These drawings can be generated from any viewing position and from any distance relative to the body. Such a system lends itself easily to aircraft motion studies since these vehicles translate along and rotate about all three axes of motion. However, the static drawings created by MOVIE.BYU do not adequately portray complex time dependent maneuvers, so we needed an easy to use system which would demonstrate such motions.

Using the ability to draw a body from any relative angle and distance, we attempted to develop the required system by overlaying a sequence of aircraft pictures on one sheet of paper. We call such drawings Poster Plots, and use them in aircraft mishap investigations. Quite often, two people who see the same event give differing accounts of what happened. To prevent future aircraft mishaps, Air Force engineers try to determine the sequence of events leading to a crash, then deduce what might cause such a motion. To this end, we sometimes construct a Poster Plot depicting our best guess at the mishap aircraft motions. After studying the series of pictures, a witness to the crash might be able to determine whether the sequence of drawings accurately describes what he or she has seen. If not, we might try to incorporate the critique of the witness into a new Poster Plot. By iterating through this procedure, we could sometimes arrive at a close approximation to the motions of the mishap aircraft. This method is not always successful, but it works often enough to make the procedure useful to Air Force engineers.

We wrote a subroutine called ORIENT which calculated the orientation and offset position relative to a ground fixed observer of an aircraft at some point in space and installed this subroutine into the ENFTC version of MOVIE.BYU.

Figure 8 shows a Poster Plot of the type used in mishap investigations. This Poster Plot was generated using data from an aircraft six-degree-of-freedom simulation. Each picture is accompanied by a label which gives important data concerning the motion being shown. ALT refers to altitude, VEL gives total aircraft velocity in feet per second, LF provides normal load factor at the center of gravity, PITCH lists the aircraft pitch angle, BANK expresses the roll angle, and HDG supplies the change in heading (yaw) angle from the original direction of flight. In the past, such drawings were generated by hand. Drawing an aircraft with the proper geometric perspective is nearly impossible when drawing by hand. MOVIE.BYU can draw an aircraft in a particular orientation easily, thus reducing the time and effort required to construct each sketch to nearly zero. The geometry model had to be arranged in the orientation described in Section 2 of this report. MOVIE.BYU writes questions to the terminal screen as prompts to the user, and ORIENT suppressed this output with a logical variable which prevents the WRITE statement from operating. The only exception was a prompt which asked for the position of the ground fixed observer. ORIENT used the MOVIE.BYU RESTORE command, which removes all previous rotations of each part of the model, and an EXPLODE command of parts 1 through 20 to the origin, which returns all parts of their initial position, before each drawing to reinitialize the geometry model. These commands put the model in the orientation shown in Figure 1. ORIENT used the MOVIE.BYU ROTATE command, which pivots the model about the fixed MOVIE.BYU axes, to orient the aircraft.

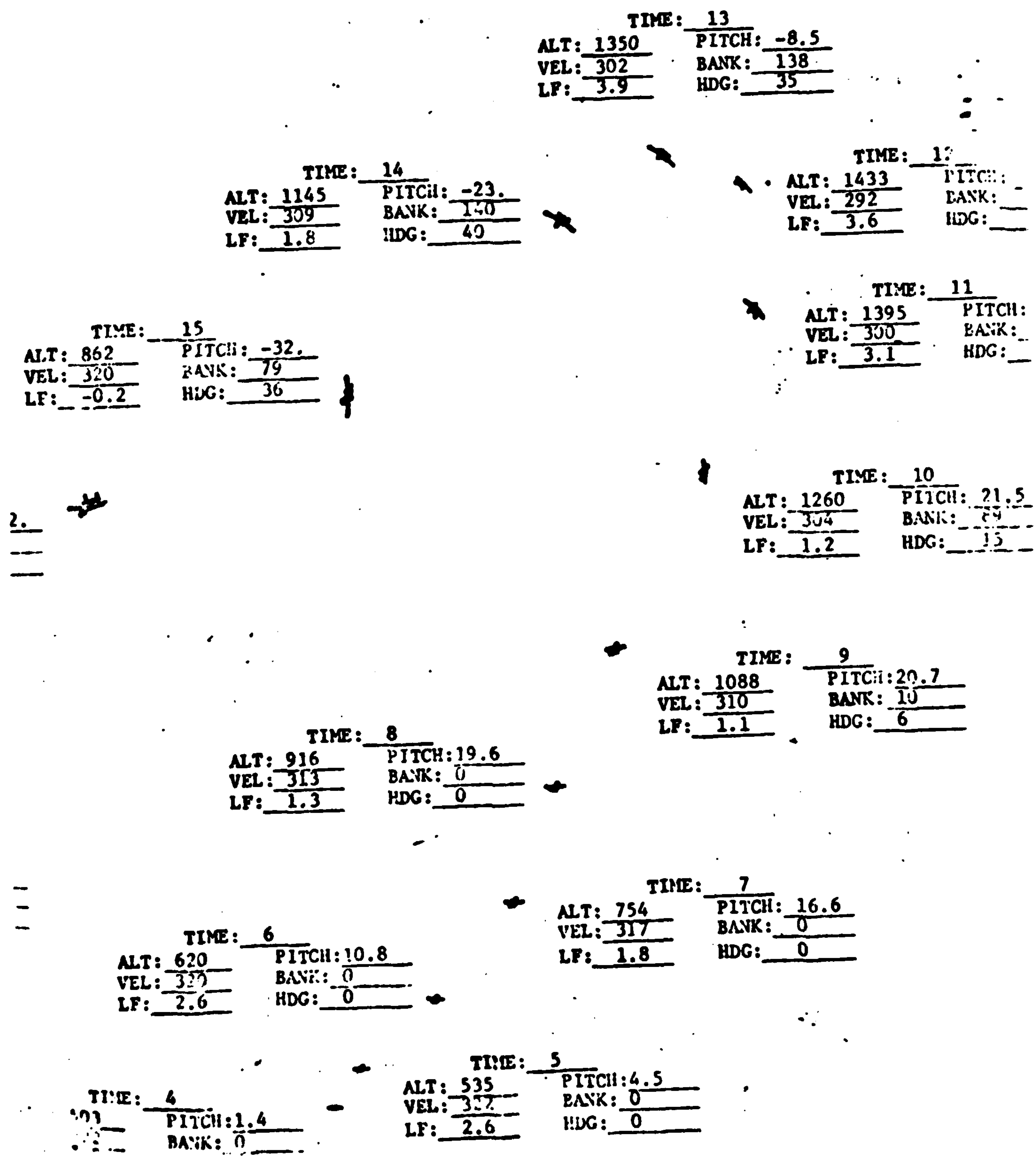


FIGURE 8

POSTER PLOT RE-CREATION

The first two rotations, 90 degrees about X and 90 degrees about Y, positioned the aircraft with wings level and the nose pointed away from the observer. The subroutine next rotated the aircraft into the proper orientation with an Euler rotation of psi-theta-phi (yaw-pitch-roll, z-y-x). It read the angles from the trajectory file. Using the observer position specified by the user and the X, Y, and altitude aircraft positions given in the trajectory file, ORIENT calculated the location of the aircraft in space with respect to the observer. ORIENT could not use EXPLODE to position the aircraft at its proper position because the data contained in the trajectory file, as well as the observer position given by the user, were given as coordinates in the fixed inertial axes, which do not rotate or translate with the aircraft. EXPLODE moves the model by placing it at a coordinate in the model body axis system, which rotates, but does not translate, with the body. Reference 2 contains a matrix which uses the Euler angle rotations psi, theta, and phi (yaw, pitch, and roll) to convert body axis linear velocities into inertial axis linear velocities. The derivation of this matrix is described in Reference 2, and is too lengthy to treat in-depth here. Briefly, this matrix calculates the projection of each body axis velocity onto each of the three inertial axes. The inverse of this matrix, which conveniently equals its transpose, can be used to convert inertial axis linear displacements into displacements along a set of axes with the same origin but which rotate to remain parallel with the aircraft body axes. ORIENT used this new matrix to convert the X, Y, and altitude offset between the observer and the aircraft into the appropriate rotational axis displacements with the observer at the origin. This matrix provides the proper position in space of the vehicle in its own axis system. The subroutine then invoked EXPLODE to position the aircraft at these coordinates. For the aircraft at these coordinates to be positioned correctly relative to the observer, the observer must be at the origin of the model. Using the MOVIE.BYU DISTANCE command, ORIENT placed the observer at the origin by setting the distance between the viewing position and observer equal to zero. This left the model completely arranged and ORIENT invoked the MOVIE.BYU VIEW command to draw the picture. It suppressed the statements which invoke the PLOT-10 commands to erase and initialize the page prior to drawing on the terminal screen. When VIEW finished drawing the picture, ORIENT read the next set of trajectory parameters and began again. Since the page erase commands had been suppressed, the pictures were drawn on the same "page." The subroutine ended after reading all the trajectory file data, and returned to normal MOVIE.BYU activity. A hard copy of the screen gave the user a Poster Plot. Poster Plots, while an improvement over earlier methods of data presentation gave no feel to the viewer for the speeds and times involved in the motions. The pictures making up the Poster Plot of Figure 8 are drawn for each second of flight time, but the only indication of this is given by the data labels. The viewer cannot just watch the motion develop, but must put some imagination into studying the Poster Plot. Furthermore, Poster Plots can only show motions as seen by a stationary observer with a fixed field of view. We often deal with pilots who observe an aircraft motion while flying a different aircraft. Such views cannot be recreated by a Poster Plot, because the viewing position is moving. Poster Plots were not sufficient for our needs.

We solved these problems with a COMP80. This machine, a high resolution callographic hard copy device manufactured by Information International, can generate microfiche, or 16mm moves. It uses a 9-track binary tape as input. Our version of MOVIE.BYU uses Tektronix PLOT-10 software as its primary output interface, so we wrote a software package that emulates low level Tektronix subroutine calls. These parallel subroutines convert the vector output of MOVIE.BYU into the appropriate hexadecimal numbers for the COMP80. For a software which does not use hexadecimal numbers, the interface can be changed to use integers. Since the tape bit pattern is identical for hexadecimals or integers, the interface should be written to use whatever type of numbers work best with the system being used. After writing the converted output onto a 9-track tape with a density of 1600 bits per inch, the MOVIE.BYU output could be entered into the COMP80. The Electronic Printing Branch of the 2750th Air Base Wing at Wright-Patterson Air Force Base owns and operates a COMP80, and we access this machine to create movies of aircraft motions. In essence, we took advantage of hardware and software already owned by the Air Force to create the capability of generating animations on paper by using Tektronix output devices or animated movies by using a COMP80.

CARTOONE grew out of the subroutine ORIENT and the interface with the COMP80. One necessary modification involved the observer. Someone watching a moving body does not maintain a constant angle of view and watch whatever happens to enter this field of view. An actual observer turns his head to keep what he is watching centered in this field of view. The MOVIE.BYU observer has a fixed field of view, always looking along the MOVIE.BYU negative Z axis, so we changed ORIENT to center the aircraft within this field of view. A body in space can be offset from an observer's line of sight in both the horizontal and vertical planes, so ORIENT uses two rotations to bring the model to the center of the observer field of view. ORIENT calculates these two angles in the following manner:

$$AZIM = TAN^{-1} \left(\frac{Y_A - Y_O}{X_A - X_O}\right)$$

$$ELEV = TAN^{-1} \left(\frac{H_A - H_O}{GDIS}\right)$$

$$GDIS = ((X_A - X_O)^2 + (Y_A - Y_O)^2)^{1/2}$$

where

AZIM = azimuth angle (positive to the right)

ELEV = elevator angle (positive down)

GDIS = ground distance between observer and aircraft

X_A = X position of aircraft center of rotation

X_O = X position of observer

Y_A = Y position of aircraft center of rotation

Y_O = Y position of observer

H_A = altitude of aircraft center of rotation

H_O = altitude of observer

The azimuth angle specifies the angle in the horizontal plane between the MOVIE.BYU observer's line of sight and a straight line between the observer position and the aircraft position. The elevation angle is the angle in the vertical plane between the observer direction of view and a straight line between the observer and the aircraft. ORIENT uses two MOVIE.BYU rotate commands to center the aircraft within the observer's field of view. The first command rotates the model about the MOVIE.BYU fixed Y axis through an amount equal to the azimuth angle, thus negating the horizontal offset of the model from the observer line of sight. The second command rotates the model about the MOVIE.BYU fixed X axis through an amount equal to the elevation angle. This cancels the vertical offset of the model from the observer line of sight. Each picture of the primary vehicle is drawn in the geometric center of the view window. The centering commands do not affect the orientation of the aircraft as seen by the observer, just as turning his head to center the aircraft in his view does not change the relative orientation of the vehicle seen by a real world observer.

The centering commands in ORIENT destroyed the automatic Poster Plot capability which depended on pictures moving across a page. To restore this ability, we constructed a series of subroutines to help control the work flow of an animation. These subroutines determine whether pictures are to be overlayed on one page or centered in the field of view on successive pages. We refer to these subroutines under the general title of View Mode subroutines because they determine how the motion is perceived. These subroutines read the trajectory data and supply this information, along with the observer position to ORIENT. The new version of ORIENT uses this data to arrange the model and center it. ORIENT then ends, returning to the appropriate View Mode subroutine. For Poster Plots, the subroutine reverses the two centering rotations, and draws the picture. For animations, the subroutine re-initializes the page and draws the picture. The whole cycle repeats until the trajectory file is completed. As we added view modes, the method of using separate subroutines rather than one large subroutine which does everything proved to be the easiest and most efficient method of adding new features to our system.

At this point in the development of CARTOONE, we could either create a Poster Plot or, using the COMP80 interface, show a motion on film from the view of a fixed position observer. These early films demonstrated the difficulty of providing a perception of three-dimensional motion on a two-dimensional screen. In a movie, motion of a body is perceived as a series of changes relative to a non-moving reference. Rotations are usually easily discernible, since the orientation obviously changes. The rotation of a sphere might not be obvious, but the rotation of any irregularly shaped object can be plainly seen. Translations are not as obvious in our movies because each picture appears in the same position on the screen. The motion of the object towards or away from the viewer presents no problem because it appears to grow larger or smaller as the relative distance between the observer and the aircraft changes. Figure 9 shows several frames from an animation of a MiG-21 Fishbed J flying straight at the observer. There is little doubt about the motion. Lateral motions are more difficult to portray, especially if the distance between the observer and the body does not change appreciably. Any changes in the observed aircraft orientation except those of apparent size may be interpreted by the audience as rotations rather than translations if there is no reference to suggest motion. Of course there is actually no motion at all, merely two-dimensional drawings of an object, but animated films can create an illusion of a solid body in motion. To provide the sensation of translation, we use a terrain model as a fixed reference. Figure 10 shows three frames from an animation of a MiG-21 moving in a circle around the observer. Nothing in these drawings indicate motion. Figure 11 shows the same motions, but with a grid of squares representing the ground. The differing positions of the vehicle relative to the grid gives the appearance of an aircraft moving over a fixed reference.

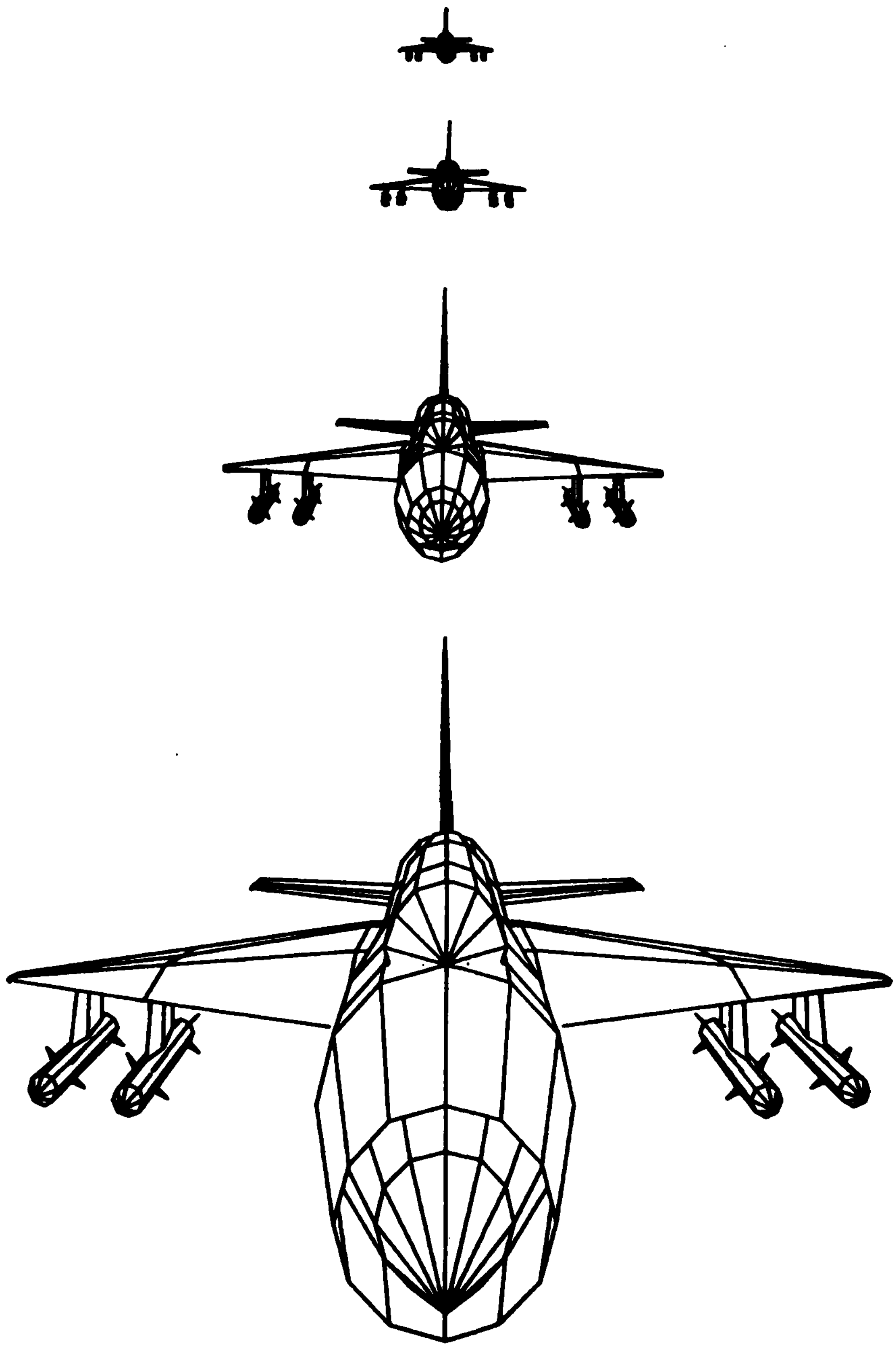


FIGURE 9
AIRCRAFT FLYING TOWARDS OBSERVER

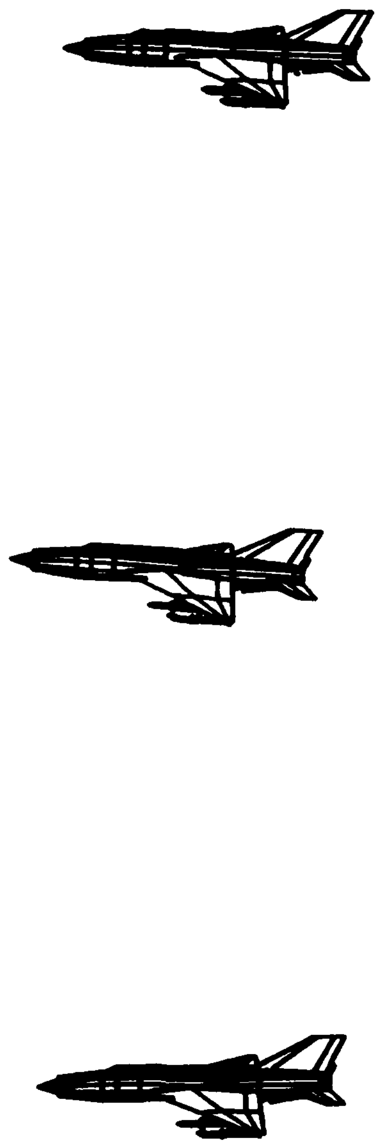


FIGURE 10
AIRCRAFT FLYING AROUND OBSERVER

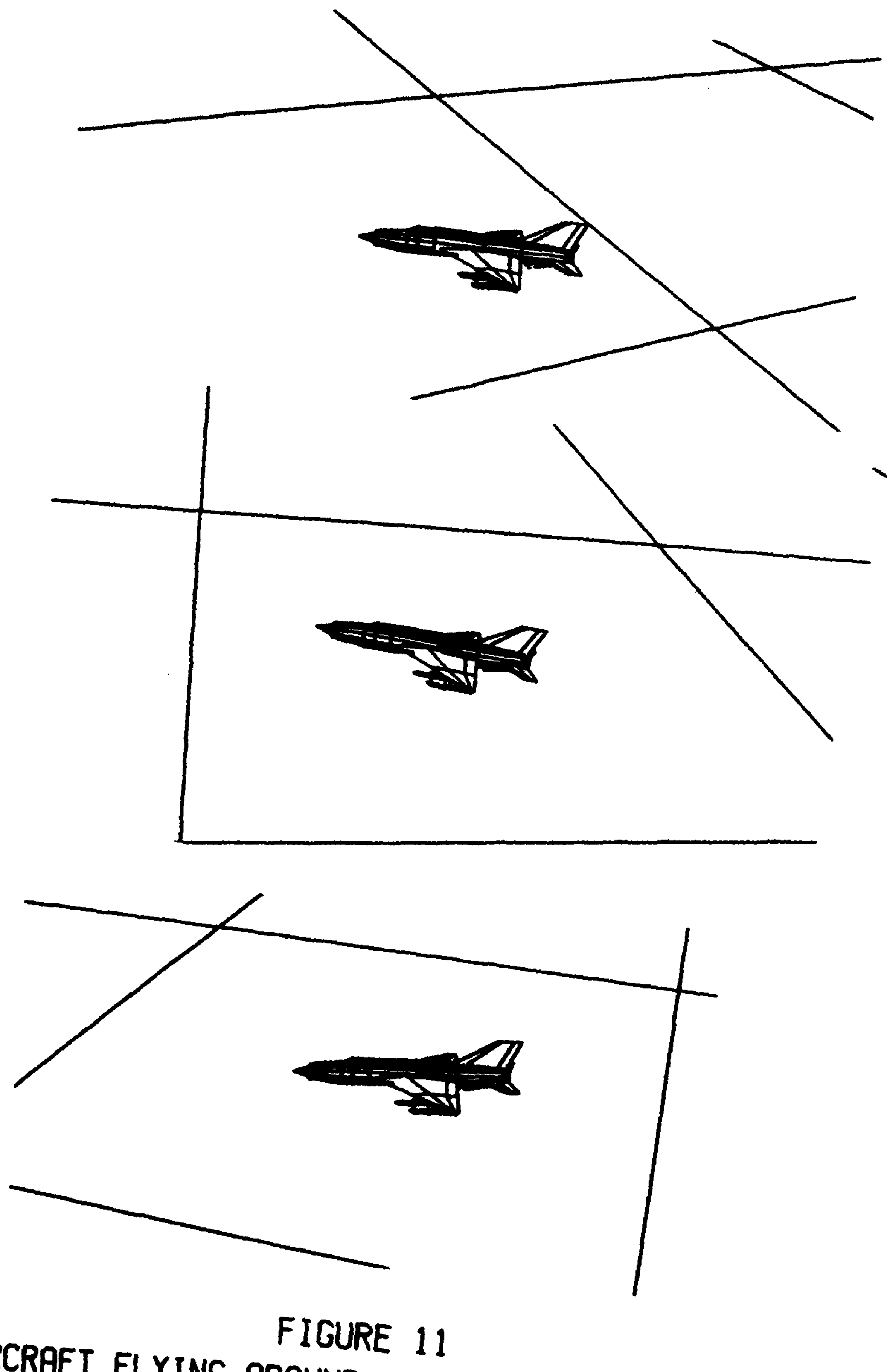


FIGURE 11
AIRCRAFT FLYING AROUND OBSERVER-WITH TERRAIN

The terrain model can be as complicated as necessary, but some fixed reference, even a grid of squares, must be present to suggest three-dimensional translation.

We modified ORIENT to include the capability of positioning the ground model. ORIENT needs the part limits of the terrain as an additional set of user input. The subroutine uses the MOVIE.BYU IMMUNE command to prevent the terrain from rotating during orientation of the moving bodies. It then takes the algebraic inverse (changes the sign) of the observer coordinates, and uses EXPLODE to position the terrain at this location. The differing positions of the aircraft relative to this terrain creates an illusion of three-dimensional motion of the vehicle. ORIENT removes the IMMUNE command before centering the vehicle, and the rotation of the terrain due to the centering commands creates the illusion of an observer turning his head to keep the aircraft in view.

Until the COMP80 interface was developed, we used a Tektronix 4014 as our primary output device. We still use this device for many purposes, including some animations. A movie is not always necessary. A series of pictures on successive sheets of paper are often adequate to describe a motion. As originally written, ORIENT overlayed pictures on one page until finished with all trajectory data, or finished drawing one picture then initialized the page for the next picture. ORIENT did not pause between pictures. An animation draws successive pictures on separate pages, and we had to modify our system to allow us to generate hard copies of the Tektronix screen for animations done at the terminal. To provide the necessary capabilities, we created a subroutine called COMB, an abbreviation for combination, which provides a combination of capabilities to the user of a

Tektronix 4014. MOVIE.BYU reads user inputs through a command unit specified by the variable ICOMM. This variable is initially set to Unit 5, which must be assigned to the terminal input file. ORIENT resets ICOMM to Unit 16 and sends all its MOVIE.BYU commands to this unit. When ORIENT finishes drawing a picture, it invokes CONB. If output is being sent to a magnetic tape for input to a COMP80, COMB re-initializes the page and invokes the proper subroutine to generate the next frame. If the user is creating a Poster Plot, the page is not re-initialized and the next picture is drawn. If the output device is a Tektronix 4014 and an animation is being generated, COMB resets the command unit variable ICOMM to 5, and invokes the PAUSE feature of COMB. PAUSE prompts the user for input, and reads a character from the terminal screen. Nothing happens until the user returns a character, giving the user an opportunity to generate a hard copy of the screen. If the user returns a blank, COMB invokes the proper View Mode subroutine which generates the next picture of the animation. If the user returns any character other than a blank, the animation stops and MOVIE.BYU activity resumes. This feature makes it possible to discontinue an animation without terminating execution of the program. The user can restart the animation with different options by going through the initialization procedures and choosing the appropriate run time options. To resume the animation at the point it ended, the user enters the command "COMB", and when COMB prompts the user for input, the user should return a blank. COMB invokes the appropriate View Mode subroutine and the animation continues. ORIENT reads trajectory data through Unit 13, a unit that MOVIE.BYU does not use. Since ORIENT first restores the model to its original configuration, the animation can proceed at the point that the user returned to MOVIE.BYU.

COMB provides other useful features to the user besides providing the abilities to stop or continue animations and a chance to generate pictures of the screen. Another option we felt would be useful was the ability to move ahead through the trajectory file to a time interval of interest without drawing each intervening picture. The SKIP feature of COMB provides this capability. If the user invokes SKIP, COMB asks the user for the time of interest, and reads the user's reply with a free format READ statement. COMB reads successive sets of trajectory data until it finds an interval with time greater than or equal to that specified by the user, then prompts the user for another command. If it reaches the end of the trajectory file without locating the required time step, it returns to the time step from which it began the search, thus returning the trajectory file to the condition it was in before the search began. It does this by storing the desired time step under the variable TWT, rewinding Unit 13 if an end of file is encountered without finding the time of interest, resetting TWT to the original time step, then reading successive trajectory intervals until it reaches the original time step. We often find it useful to determine the current values of the trajectory file parameters, and the DATA option provides this capability. It causes COMB to write the current value of the seven primary trajectory file parameters and the current location of the observer on the upper left corner of the terminal screen. The user can invoke the REWIND option to start the animation over again with the same run time options. This option causes COMB to rewind Unit 13, through which ORIENT reads the trajectory data. In many cases, we find it useful to input trajectory data by hand, and the DEBUG option provides this capability. Our system reads trajectory file data through a unit specified by the variable

IDATA. DEBUG causes COMB to change the value of IDATA from 13, the default trajectory input unit, to 5, the terminal input unit, allowing the user to input data manually. The user is prompted for trajectory input with a request for data. Invoking DEBUG a second time changes IDATA back to 13. Our last addition to COMB was a HELP feature, which lists and briefly describes each COMB option. The options RETURN, DATA, REWIND, and DEBUG all cause COMB to end and normal MOVIE.BYU activity can continue. We changed ORIENT to make it invoke the data option of COMB prior to pausing whenever the output device is 4014.

Our system consisted at this point of four subroutines (ORIENT, COMB, and the two View Modes, FIXED (for animations) and MURAL (for Poster Plots)), two hardware interfaces (PLOT-10 and COMP80), and several user options. To simplify the user's task, we built another subroutine to obtain user input, do I/O unit assignments, and begin the animation. We called this subroutine CART, short for CARTOONE. We also refer to the entire package of subroutines and hardware interfaces by the name CARTOONE. CART asks the user which option, such as output device, should be used. It supplies this information to the appropriate subroutines. We added other options to the CARTOONE system by changing CART to obtain the necessary information from the user, and either modified ORIENT or the View Mode subroutines to accomplish the task or built extra subroutines to do the task if that method was more efficient. The only parts of the system which the user can directly access from MOVIE. BYU are COMB and CART, both of which are allowable commands of our enhanced version of MOVIE.BYU. The rest of this section describes the development of options and features we added to CARTOONE.

Using the MOVIE.BYU DISTANCE command, we added the Zoom Factor option to CARTOONE. As the distance between an object and an observer increases, the object becomes increasingly difficult for the observer to see. Zoom Factor provides a close-up look at a motion being reproduced. The DISTANCE command changes the distance between the MOVIE.BYU viewing position and the origin of the geometry model. Nominally, the observer is placed at the model origin. The Zoom Factor feature is made up of the following two consecutive statements in ORIENT:

$$DIST1 = ((H_A - H_O)^2 + GDIS^2)^{1/2}$$

$$DIST = \left(\frac{1}{ZOOM} - 1\right) * DIST1$$

DIST1 = actual distance between observer and aircraft

DIST = perceived distance between observer and aircraft

ZOOM = Zoom Factor (supplied by user to CART)

A Zoom Factor of one leads to the nominal case by setting the distance between viewing position and the model origin equal to zero. Zoom Factors greater than one reduce the relative distance between observer and object, while Zoom Factors less than one increase the relative distance. CART will not accept a Zoom Factor less than or equal to zero. The apparent orientation of the body remains the same, but it appears bigger or smaller.

Fixing the observer limited the practical applicability of CARTOONE. Watching a motion from a fixed position distorts the appearance of the motion since the apparent orientation of the object depends heavily on the geometric relationship between the observer and the aircraft. Figure 12 shows a KC-135 Stratotanker as seen by three different observers. The difference between the observed orientation of the aircraft and its actual

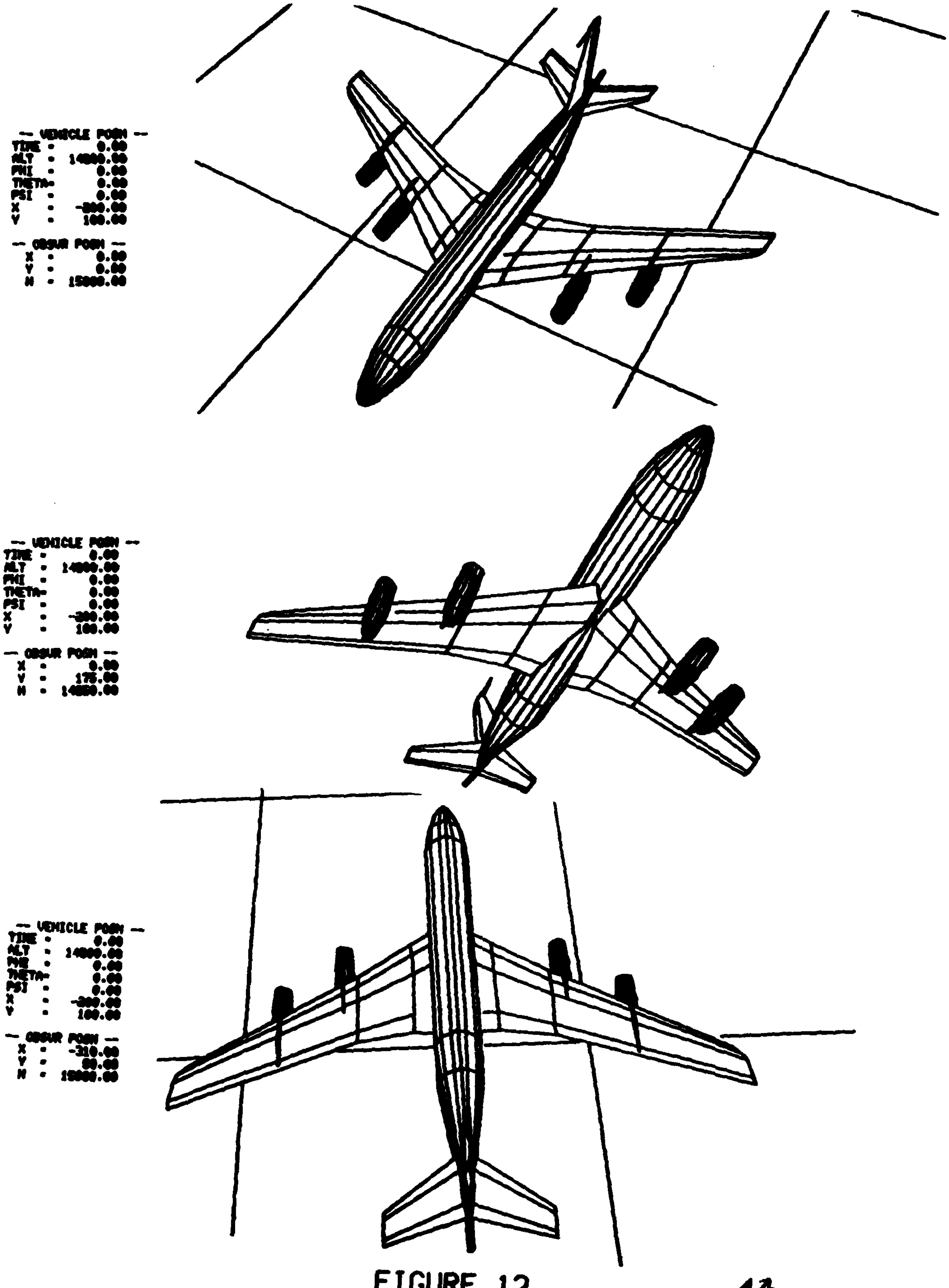


FIGURE 12

EFFECT OF OBSERVER MOTION

orientation is referred to as a parallax error, or simply parallax. For a fixed observer, parallax changes as the aircraft moves. To accurately observe the true motion of a moving body, the witness must translate with the body, thus maintaining a constant parallax. We developed the Wingman mode (with view mode subroutine WNGMAN) to allow the observer and the aircraft to translate together. The user specifies an initial observer position relative to the aircraft, and the observer position for each subsequent time step is calculated in the following manner:

$$X_O = X_A + RX$$

$$Y_O = Y_A + RY$$

$$H_O = H_A + RH$$

where

RX = relative distance in X direction between aircraft and observer (positive forward).

RY = relative distance in Y direction between aircraft and observer (positive right)

RH = relative difference in altitude between aircraft and observer (positive up)

ORIENT positions the terrain at its proper position relative to the observer for each frame of the animation. The observer stays at the model origin, so the Wingman mode actually causes the terrain to move beneath the observer.

WNGMAN was the third View Mode subroutine of our system. Each of these subroutines determine the aircraft and observer locations for each time step, and supplies ORIENT with this information. The user supplies the View Mode to CART which then invokes the appropriate subroutine to start the animation. The View Mode is stored in a common block under the variable OBS. After ORIENT arranges the model, the View Mode subroutine draws the picture and invokes COMB. When the user commands COMB to restart the animation, it uses the variable OBS to invoke the proper View Mode subroutine. If the user wants to re-initialize an option, he must go back through CART.

So far, the user could choose between Poster Plots or two animated View Modes, Fixed Position and Wingman. As useful as these are, we needed other View Modes. Nonstationary viewers usually do not maintain a constant relative distance from an aircraft, but fly along an independent flight path. One such group we often encounter are pilots who had been flying in formation with (i.e., parallel to) an aircraft when it began some maneuver. For rapidly developing motions, the chase pilot might not follow it through the maneuver, but continue along the original flight path. To provide such a view, we developed the Chase Plane View Mode, with the related subroutine CHASE. When the user requests the Chase Plane mode, CART asks for the initial position of the observer relative to the aircraft along each axis just as it does for the Wingman mode. CHASE uses the aircraft trajectory to calculate the "components" of the "initial velocity vector" of the aircraft. It reads the first two intervals of the trajectory file and computes three "velocity vector components" in this way:

$$V_X = \frac{X_2 - X_1}{T_2 - T_1}$$

$$V_y = \frac{Y_2 - Y_1}{T_2 - T_1}$$

$$V_Z = \frac{H_2 - H_1}{T_2 - T_1}$$

V_X = velocity in X-direction (forward)

V_y = velocity in Y-direction (side)

V_Z = velocity in Z-direction (vertical)

X_1 = aircraft X position (first interval)

X_2 = aircraft X position (second interval)

Y_1 = aircraft Y position (first interval)

Y_2 = aircraft Y position (second interval)

H_1 = aircraft altitude (first interval)

H_2 = aircraft altitude (second interval)

T_1 = time of first interval, usually zero

T_2 = time of second interval

The positive Z axis for an aircraft is down, so the Z velocity has a negative sign (increasing altitude corresponds to a negative Z velocity). After calculating these "velocities," CHASE rewinds IDATA, the trajectory data unit. Each subsequent observer position is calculated in the following manner:

$X_o = X_1 + RX + V_X * (T-T_1)$

$Y_o = Y_1 + RY + V_y * (T-T_1)$

$H_o = H_1 + RH - V_Z * (T-T_1)$

RX = relative forward distance of observer with respect to aircraft at time $T = T_1$ (positive forward)

RY = relative side distance of observer with respect to aircraft at time $T = T_1$ (positive right)

RH = relative altitude of observer with respect to aircraft at time $T = T_1$ (positive up)

T = current time

The observer moves in a straight line, but does not rotate. The aircraft is centered in the view window.

The Chase Plane view did increase our capability but we needed another mode to model a pilot flying along a non-constant trajectory while watching another aircraft. For this purpose, we built the Independent Vehicle mode, with subroutine INDEP. To use this mode, the trajectory file must contain time histories describing the motions of both the observer and the aircraft. The first seven parameters of the file are the first trajectory interval of the observer, and are followed by the first interval of the aircraft trajectory, and so on. The view as seen by an Independent Vehicle observer centers on the primary vehicle and does not rotate, even though dummy data must be supplied for the viewer rotations. INDEP supplies ORIENT with the X, Y, and altitude positions of the observer as well as the seven trajectory parameters of the aircraft.

The final View Mode we have created thus far is the view of the pilot flying the aircraft itself. This mode, called pilot eye, has one major difference from the others, the aircraft itself is not drawn. This View Mode does not access ORIENT. The subroutine PILOT arranges the model prior to drawing the picture and invoking COMB. PILOT uses the MOVIE.BYU PARTS command to withhold the aircraft from view. PILOT next reads the trajectory data, and positions the ground model in the same way as ORIENT. Note that the observer position is the same as the aircraft position, so only the terrain needs to be positioned. The subroutine calculates the algebraic inverse (changes the sign) of each aircraft rotation, and performs an Euler rotation of phi-theta-psi (roll-pitch-yaw, X-Y-Z) with these inverted angles on the ground model. Pilot uses the MOVIE.BYU FIELD command to change the pilot angle of view from its nominal value of 28 to 90. This provides better peripheral vision than normally available to the MOVIE.BYU viewer. PILOT places the observer at the origin of the geometry model by setting the distance between observer and origin to zero, draws the picture, and invokes COMB. PILOT causes the ground to translate past and rotate around the observer, creating the illusion of a view from the cockpit. The observer looks along the aircraft positive X-axis (out the nose).

The five View Modes, along with Poster Plots, give the user the capability to portray any kind of motion of one vehicle as seen by any kind of observer. To expand the usefulness of CARTOONE, we added the capability of manipulating more than one vehicle on the screen, and call this feature Multiple Independent Vehicles. We modified ORIENT and PILOT to arrange successive bodies with data taken from successive blocks of numbers from the trajectory file. These bodies must be separate parts of the geometry model, and the user specifies the part limits of each body to CART. MOVIE.BYU limits models to 19 separate parts, so CARTOONE is also limited to 19 parts. One part or one set of parts must be the terrain model if a terrain is to be used. The terrain does not need a time history, its position relative to the observer is determined by the observer motion. The view remains centered on the primary vehicle for all view modes except Pilot Eye and Mural (Poster Plots). CARTOONE assumes that the primary vehicle is governed by the first set of parameters given in the trajectory file, unless the Independent Vehicle view mode is used. In this case, the observer trajectory file is first and the primary vehicle trajectory is second. The structure of the trajectory file should be as follows:

a. Seven parameters for independent vehicle (optional)

b. Seven parameters for primary vehicle

c. Seven parameters for each successive body (optional)

This pattern repeats for each time step of the trajectory. If CARTOONE reaches the end of the file, or if it encounters an error while reading from the trajectories, CARTOONE ends. While it is theoretically possible to show as many as 19 separate bodies moving along independent flight paths on the screen, it is rarely feasible to show more than three at a time. To keep a good percentage of the action on the screen, the observer has to move farther away for each body added to the total since the view always centers on the primary aircraft. MOVIE.BYU clipping routines reduce the amount shown at a time unless the observer distance is great enough to include all the bodies in its field of view. Furthermore, very complex models may encounter problems with the Watkins Hidden Line Algorithm not having sufficient scratch space to draw the model.

The DATA option of COMB causes trajectory information to be printed on the screen for each "frame" of the animation when the output device is a Tektronix 4014, making quantitative data automatically available to the user. We had no parallel capability for COMP80 animations. We felt that this was a serious deficiency. Putting information on the screen of a Tektronix 4014 is as simple as writing the required information to an output unit which writes to the terminal. To print letters on COMP80 film, we had to convert each letter into an appropriate hexidecimal symbol. The COMP80 treats a reserved set of such symbols, called "fonts", as characters, and the user can use these fonts to write character strings onto the film. We use these fonts to provide data to the viewer.

The first method we investigated was projecting data on the screen during an animation. Although this method works well for Tektronix animations, it tends to distract the attention of viewers during a movie. The viewer may have to replay the film several times to fully comprehend what the movie was telling him, thus reducing the effectiveness of the movie.

Anyone who has watched a sporting event on television in which the athlete is being timed has been distracted by the flickering of the digital clock on the screen. This clock draws the attention of the viewer away from the action. Our movies are primarily intended to display aircraft motions, and anything which distracts the attention of the audience away from the motion defeats the purpose of the movie. Therefore, we removed this feature from CARTOONE. The capability can be restored should the need arise, but as an implicit CARTOONE option, it is unavailable.

There is one exception to this rule. In many cases, it is essential to know the pilot actions during an aircraft motion. This need led to the construction of the Control Block feature, which presents pilot control inputs graphically on the screen. Presenting these inputs graphically, rather than numerically, reduces distraction since a glance is sufficient to tell the viewer what the pilot actions are. The "control block" consists of a large cross and a horizontal bar beneath the cross, along with two small indicators, or pippers. Figure 13 shows an example of a control block. The cross represents the pilot control stick. The vertical bar of the cross shows the longitudinal position of the stick, while the horizontal bar of the cross depicts the lateral stick motion. The place at which the two lines cross represents the centered (or undeflected, or neutral) stick position. The horizontal line below the cross represents the pilot rudder pedal motions. When the user invokes this option, CARTOONE draws the control block in the lower left corner of the screen for each frame of the animation. If the user intends to use this option, he must supply three additional parameters for the primary vehicle in the trajectory file, the longitudinal stick position, lateral stick position, and rudder pedal position.

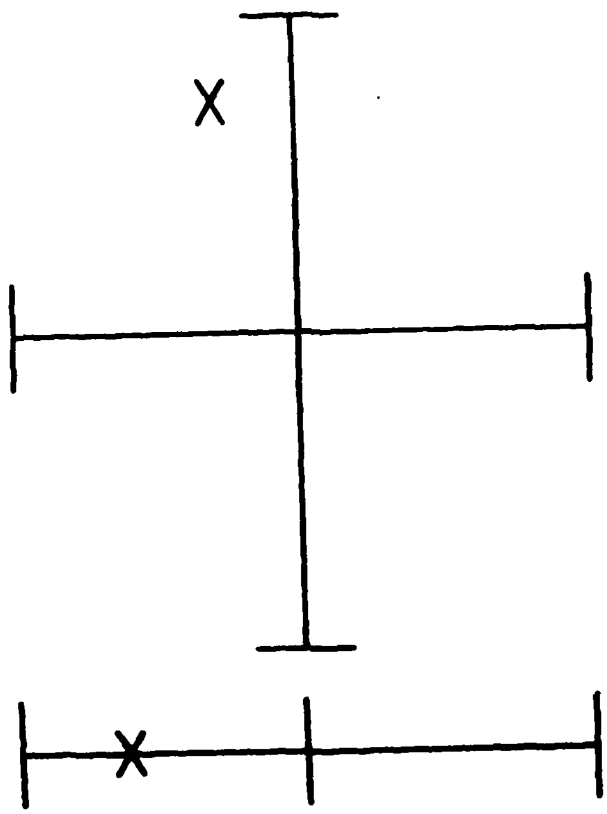


FIGURE 13
CONTROL BLOCK

CARTOONE uses this data to place the control block pippers in the proper position to reflect the pilot control motions. These three additional parameters must be present for each interval of the primary vehicle trajectory. Control Block does not use absolute control motions to position the pippers since maximum control deflections differ between aircraft. Instead, CARTOONE reads control deflections as fractions of maximum. For example, if a stick can be moved aft five inches, a motion of three inches would be entered in the trajectory file as 0.6. CARTOONE would place the pipper 60% of the way down the vertical bar of the cross from the neutral (i.e., centered) position. This makes the Control Block option generic in nature. The statements which draw the control block are embedded in the MOVIE.BYU subroutine BGNFRM which initializes the page and draws the border around the view window. This subroutine accesses PLOT-10 subroutines, and parallel COMP80 subroutines, to initialize the page and draw the lines. When the user invokes CART, one of the user replies trips a logical condition which indicates whether or not a Control Block should be drawn. BGNFRM uses this logical condition to determine whether or not to draw the figure.

We still faced the problem of providing quantitative trajectory data to the viewer, and a parallel problem of introducing animations to describe what is to be shown. We use the COMP80 fonts to create Title Pages. We use these pages to communicate information directly to the audience. Since they precede the animations, Title Pages do not distract the viewer's attention from the vehicle motions. We experimented with various letter sizes and page lengths to determine the best format for a Title Page. We settled on a page consisting of four lines of approximately 20 characters each. The lines of letters are separated from each other by a blank line of the same size as the letters. When the user specifies output device COMP, he may choose to precede the animation with one or more Title Pages. There is no upper limit to the number of Title Pages which the user can generate. The user enters each line of each page, and CART reads it with a "20A1" format. CART analyzes each character and translates it into the proper hexidecimal font. If the user enters an undefined character, one which is not represented by a font, the line is rejected and the user asked to re-enter the entire line. CART writes each Title Page onto the tape 144 times, which creates six seconds of film at 24 frames per second.

Each step in the development of CARTOONE, except the interface with the COMP80, used existing MOVIE.BYU options. We created subroutines to perform required calculations and invoke the MOVIE.BYU commands in the proper way. CARTOONE began as an outgrowth of MOVIE.BYU, and now exists as a system which uses MOVIE.BYU as an output tool. While previous output methods serve their always useful purpose, CARTOONE adds an extra capability which makes our job easier to perform.

SECTION V

FUTURE ADDITIONS

The preceding sections discussed the features of CARTOONE available at the time this report was written. We do not intend to stop at that point. We intend to increase the number of options available, making the system as useful to as many people as possible. This section describes the features we plan to add to CARTOONE in the near future.

Most 16mm projectors can only run at a speed of 24 frames per second. To use CARTOONE to generate a movie, the user must generate 24 trajectory intervals for each second of motion, and must generate 24 sets of translated MOVIE.BYU output for the COMP80. An animated movie creates an illusion of motion by rapidly showing a sequence of pictures. The pictures are a digitization of a continuous motion. If the time interval between pictures is small, no two successive pictures differ greatly enough to make the changes easily detectable. A projector displays these pictures at the same time interval they were taken, thus effectively re-creating the continuous motion. The more rapidly that a motion develops, the smaller is the time interval necessary to ensure that the digitization does not cause abrupt changes between time steps. The standard 16mm projector rate of 24 frames per second is sufficiently small to smoothly re-create almost any motion. In fact, most motions can be smoothly digitized with a larger time step. However, projecting such a sequence at 24 frames per second shows a motion that develops faster than the original motion. To accurately re-create a motion, each frame of a digitization must be shown for a length of time identical to the interval of the digitization. For example, if a motion is digitized with an interval of 12 frames per second, each picture must be shown for 1/12 of a second, rather than 1/24 of a second that a 16mm projector will show it. This can also be achieved by creating two successive frames of each trajectory interval. A similar procedure can be arranged for any required time interval greater than 24 frames per second. At present, CARTOONE generates an individual set of output for each step of the trajectory file, even if an interval is repeated. We intend to add the capability of repeating a time step without repeating all the calculations required to generate the output. This option would probably be available for both PLOT-10 and COMP80 animations, although it would be primarily intended for the generation of movies. Such a capability would reduce the time required to generate the tapes of COMP80 input if using a larger time interval is feasible, thus reducing the turn-around time for creation of movies. This feature would require CARTOONE to store in memory all the data which makes up a frame and repeat the output statements. The option would default to one picture for each trajectory interval.

We intend to modify the Independent Vehicle mode to allow the observer to rotate as well as translate. The primary vehicle would be centered on the screen. We also want to create a View Mode of an observer with a fixed angle of view. Essentially, it would be identical to Poster Plots except that the pictures would be drawn on successive frames, rather than overlayed on one page. This mode would model a truly fixed observer, one who does not turn his head to follow a particular aircraft. This mode would be of great use for an observer watching more than one vehicle. We refer to this mode as the Stationary Mode. Another View Mode would allow the user to specify velocity components for the observer, and have the observer move in a straight flight path. We call this the Linear Flight Path Mode. Another

View Mode we intend to add is a perfect wingman, one who translates and rotates with the vehicle being observed. The Wingman Mode observer translates, but does not rotate with the aircraft. This new mode would resemble the Pilot Eye Mode in that the terrain moves around the observer. The aircraft would remain in a constant relative orientation throughout the maneuver. This mode will be called the Extra Wingman Mode.

To simplify the user's task, we will add the option of varying which body of a set of Multiple Independent Vehicles is the primary vehicle. For example, if a trajectory file contains time histories for three vehicles, the user defines the part limits which determine vehicle one, vehicle two, and vehicle three. CARTOONE now centers the observer's view on vehicle one assuming it to be the primary vehicle. This new feature would allow the user to declare which vehicle is the primary vehicle, and CARTOONE will center the observer's view on this vehicle. We call this option Variable Primary Vehicle. If the user does not invoke this option, the primary vehicle would default to vehicle one.

Another way to make CARTOONE more user friendly would be to add the capability of interpolating between time steps of a trajectory file, allowing CARTOONE to vary the frame interval of animations for a given trajectory file. This would provide an implicit fast/slow motion capability. We call this future option Variable Time Step. The easiest, although not most accurate, method would be to use linear interpolation between trajectory intervals. When we implement this option, we will investigate various interpolation methods. If not invoked, CARTOONE would draw a picture for each trajectory interval.

The Control Block option presents pilot control input directly to the user, but most Air Force aircraft have automatic flight control systems which move the control surfaces also. Pilot inputs may not completely define the control deflections which cause a particular motion. To provide such information, we intend to add the Movable Control Surfaces option. The user would supply CARTOONE with the parts (or possibly with the nodes) of the geometry model which define aircraft control surfaces, and deflection information for each surface. CARTOONE would use this information to move the surfaces during an animation. Even if the vehicle being studied does not use an automatic flight control system, this option would be useful in reinforcing the information presented by the Control Block, and in demonstrating the cause-effect relationship between deflection of a control surface and the resulting motion.

A real world camera can zoom in on (or fade back on) a shot of an object. The zoom can vary. We would like to add this feature to our present Zoom Factor option. This modification might require an addition to the trajectory file parameters, or a different data file might be required. However it is done, it would add a useful feature to CARTOONE.

At present, Title Pages are displayed on 144 frames, six seconds worth of film. Some titles require longer than six seconds to be fully understood by the audience, so a more useful feature is a Variable Title Length option. The user could specify on how many successive frames a particular title should be shown. It would default to some specified value, probably 144.

For some applications, drawing a picture with the Watkins Hidden Line Algorithm of MOVIE.BYU is unnecessary or impractical. MOVIE.BYU has the VIEW command, which invokes the hidden line algorithm, and the DRAW command, which does not invoke the algorithm and draws every line in the model. We will add the capability of choosing which command CARTOONE uses to draw the pictures.

As more features occur to us, we will add to our list of future additions. The CARTOONE system has so many potential applications, we foresee many other additions. We welcome suggestions from all sources.

APPENDIX A - SAMPLE ANIMATIONS

This appendix contains several samples of how CARTOONE can be used for aerodynamic applications. The example motion, illustrated in Figures 14 through 58, shows an F-15 Eagle performing a 720-degree roll maneuver. The animation does not necessarily reflect the actual flying qualities of the F-15 or any other aircraft, and any similarity is coincidental. However, the series of pictures do demonstrate the major functions of CARTOONE. CARTOONE generated these pictures on a Tektronix 4014 graphics terminal.

CARTOONE generated the first three animations with the time history trajectory listed in Table 3. The aircraft flies straight and level for one second then rolls through 720 degrees. The aircraft finishes the maneuver in a nose low attitude, as indicated by the negative values of pitch angle, and pointed slightly to the left as indicated by the negative values of yaw angle. The "pilot" pulls back on his control stick to raise the aircraft nose and gain altitude. The trajectory is digitized at one second intervals, so the aircraft orientation changes quite a bit between pictures.

The first animation shows the maneuver from the Chase Plane Mode in Figures 14 through 24. The aircraft loses altitude during the maneuver, but the viewing position moves parallel to the initial velocity vector of the F-15 and remains at a constant altitude. As the distance between observer and aircraft builds, the motion becomes increasingly difficult to accurately observe. Changes in pitch attitude are especially difficult to detect. CARTOONE provides the trajectory data on the screen with the DATA option of COMB, making this information automatically available to the Tektronix user. This data cannot replace a good representation of the motion.

TABLE 3

TRAJECTORY FOR FIRST THREE ANIMATIONS

TIME	ALTITUDE	ROLL	PITCH	YAW	X	Y
0	1000	0	0.943	0.0	0.0	0.0
1	1000.16	0	0.963	0.0	844.494	0.0
2	993.287	142.341	-4.012	1.338	1688.86	10.868
3	932.169	306.478	-3.019	-0.921	2532.05	24.880
4	864.959	464.726	-6.947	1.294	3377.64	19.039
5	756.450	632.664	-6.798	-2.047	4221.17	25.172
6	632.659	719.856	-6.318	-1.392	5066.19	3.553
7	546.326	720.089	-0.662	-1.320	5919.65	-19.411
8	546.454	719.945	5.709	-1.659	6779.11	-42.493
9	641.938	720.000	12.198	-1.567	7631.31	-65.885
10	832.115	720.008	18.785	-1.561	8462.09	-88.565

The second animation shows the motion from the Wingman Mode in Figures 25 to 35. The initial observer position is identical, relative to the F-15, as in the Chase Plane animation. With this View Mode, the observer stays a constant distance from the vehicle. This type of view gives the viewer a better look at the total motion of the aircraft. Note how much more apparent pitch changes during the recovery become. The motion of the observer, translating but not rotating with the aircraft, does not constitute a "real world" view of the aircraft motion, especially for rapid, large amplitude motions. Chase Plane provides a more realistic view of the action, but some details are obscured or lost as the aircraft moves away from the viewer.

Figures 36 through 47 show the motion from the Fixed Position Mode. The first two pictures show the first trajectory interval with Zoom Factors of 1 and 12.5. The balance of the animation is shown with the latter zoom factor. This View Mode has been the most widely used since most observers move very little relative to an aircraft while watching it move. The major drawback of showing a motion from Fixed Position is the tendency to lose sight of the terrain since most viewers watch a motion from ground level and look upwards. If the aircraft moves close to directly above the observer, the ground moves out of sight and the reference is removed from the screen, which removes the illusion of three-dimensional motion of a solid body. Furthermore, animations shown from this view almost invariably require the use of a Zoom Factor. Figures 36 and 37 illustrate why this is so. An aircraft translates quite a bit, even during short duration maneuvers at low speeds. For example, an aircraft flying at 150 knots will travel 2500 feet in 10 seconds. A typical aircraft maneuver takes 20 to 30 seconds of flight at speeds of 250 to 300 knots, and the vehicle translates from 8500 to

15,000 feet. To show it with Fixed Position, the user will need to specify a Zoom Factor of 25 to 50 to properly show the entire maneuver.

The Independent Vehicle Mode, which allows the view position to move anywhere, could generate any of the preceding animations also. A sample of this mode is not given since it is a generic type of view mode, and can show a motion for any type of external viewing motion. All other external views can be thought of as special cases of this mode.

Table 2 contains the trajectory for the fourth animation. This time history contains control inputs, allowing the user to utilize the Control Block feature. Figures 48 through 58 show the maneuver from the Pilot Eye View Mode, with a Control Block. This mode has its greatest use in studying large amplitude coupled motions, which can be deceiving to pilots. This mode provides a way to analyze a motion from the point of view of a pilot, and can also "play back" a motion to a pilot.

Figure 59 shows a Poster Plot of the sample trajectory from the view of an observer to the right of the plane of motion. The Poster Plot observer has a 90-degree angle of view, and anything outside of the field will not be drawn on the figure. Since the vehicle covers more than 8400 feet in the X-direction during the example motion, the viewing position must be offset at least 4250 feet from the plane of motion for the entire maneuver to be drawn. Such a large lateral offset results in each drawing of the vehicle being an indistinct shape on the plot. This plot illustrates the altitude changes and the axial motion well, but does not illustrate roll motion or lateral translation well. Using a Zoom Factor to enlarge the pictures will not help because the Zoom Factor moves the observer position closer to the motion while the total angle of view is constant, and some of the motions will be lost. This problem can be circumvented by drawing the Poster Plot from a different viewing position shown in Figure 60. This plot is drawn with a Zoom Factor of 25, and the rolling motion can be seen. Because the view is from the front, the entire motion is contained within the field of view of the observer, even for a large Zoom Factor. Each picture is much more distinct than in Figure 59, and the total motion is more adequately portrayed. If the observer position is unimportant, it can be chosen appropriately to allow the use of a Zoom Factor. If not, a Poster Plot may be unsuitable for portraying the motion as seen by a particular observer.

TABLE 4

TRAJECTORY FOR FOURTH ANIMATION

TIME	ALTITUDE	ROLL	PITCH	YAW	X	Y	LATERAL	LONGITUDINAL	PEDALS
0	1000.000	0.000	0.943	0.000	0.00	0.000	0.000	0.0	0.0
1	1000.160	0.000	0.963	0.000	844.494	0.000	0.000	0.0	0.0
2	993.287	142.341	-4.012	1.338	1688.86	10.068	0.5	0.0	0.0
3	932.169	306.478	-3.019	-0.921	2532.05	24.880	0.5	0.0	0.0
4	864.959	464.726	-6.947	1.294	3377.64	19.039	0.5	0.0	0.0
5	756.450	632.664	-6.798	-2.047	4221.17	25.172	0.5	0.0	0.0
6	632.659	719.856	-6.318	-1.392	5066.19	3.553	0.00765	0.233	0.0
7	546.326	720.089	-0.662	-1.320	5919.65	-19.411	-0.176	0.233	0.0
8	546.454	719.945	5.709	-1.659	6779.11	-42.493	.00514	0.233	0.0
9	641.938	720.000	12.198	-1.567	7631.31	-65.885	.00094	0.233	0.0
10	832.115	720.008	18.785	-1.561	8462.09	-88.565	-.0011	0.233	0.0

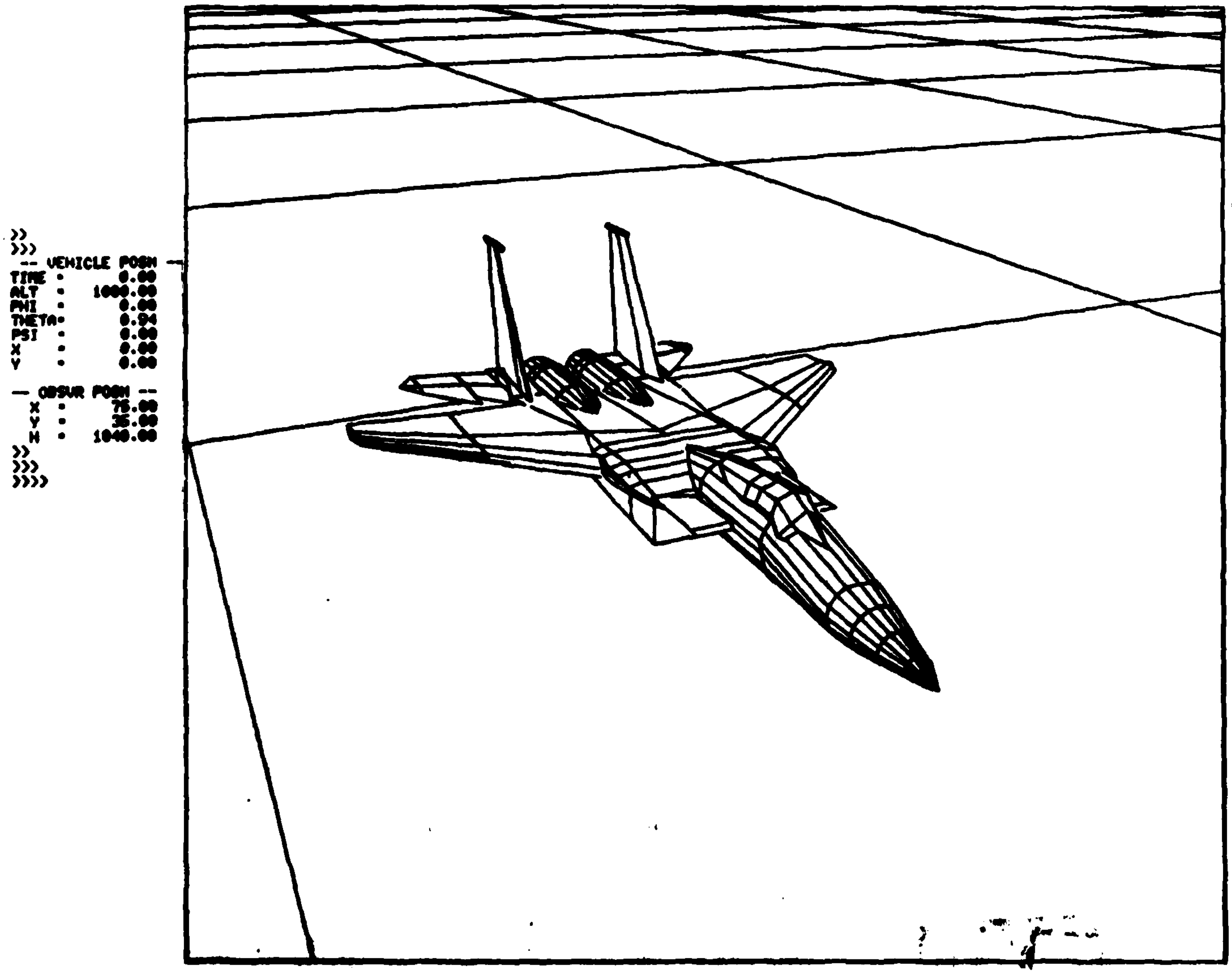


FIGURE 14
CHASE PLANE MODE - T=0

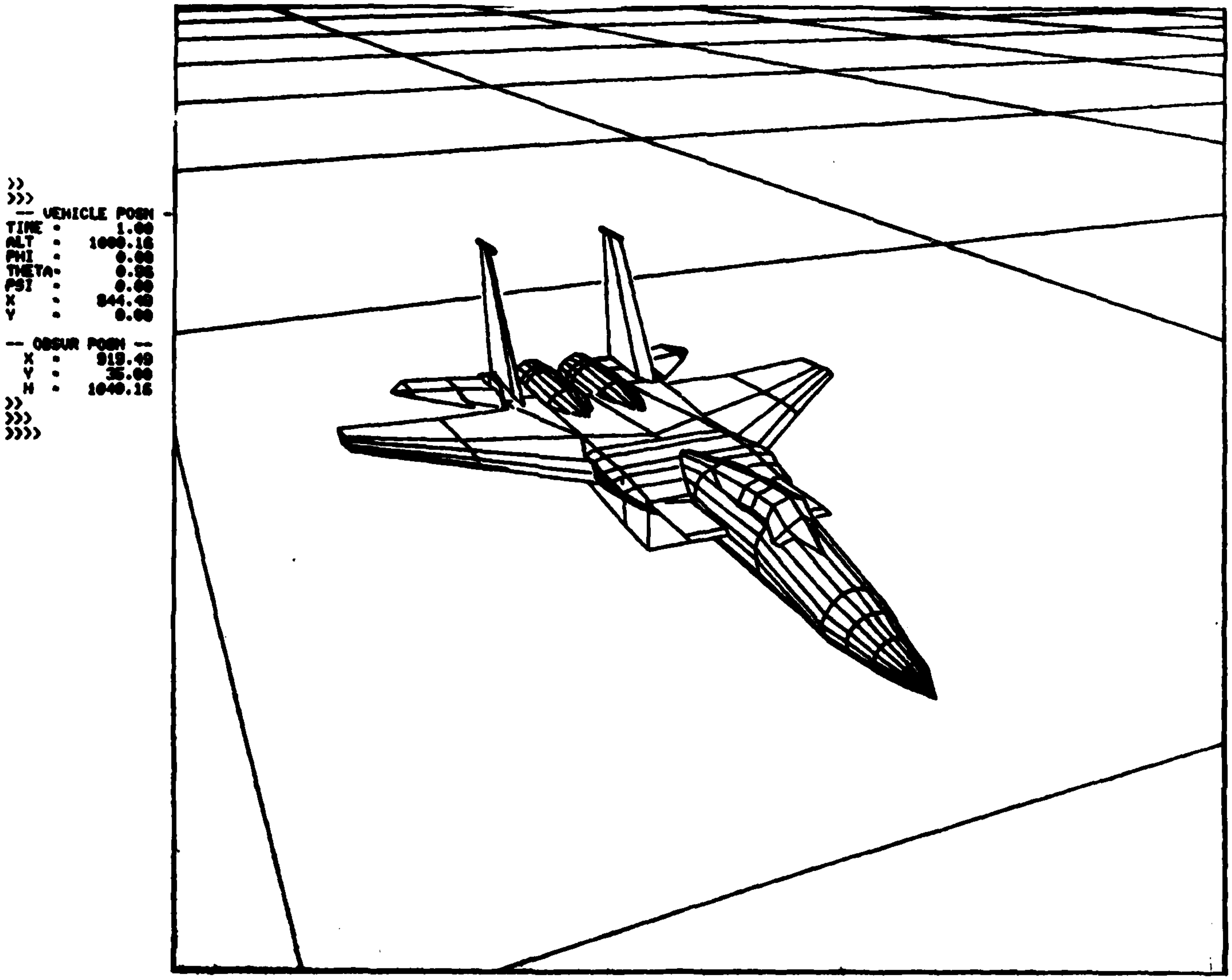


FIGURE 15
CHASE PLANE MODE - T=1

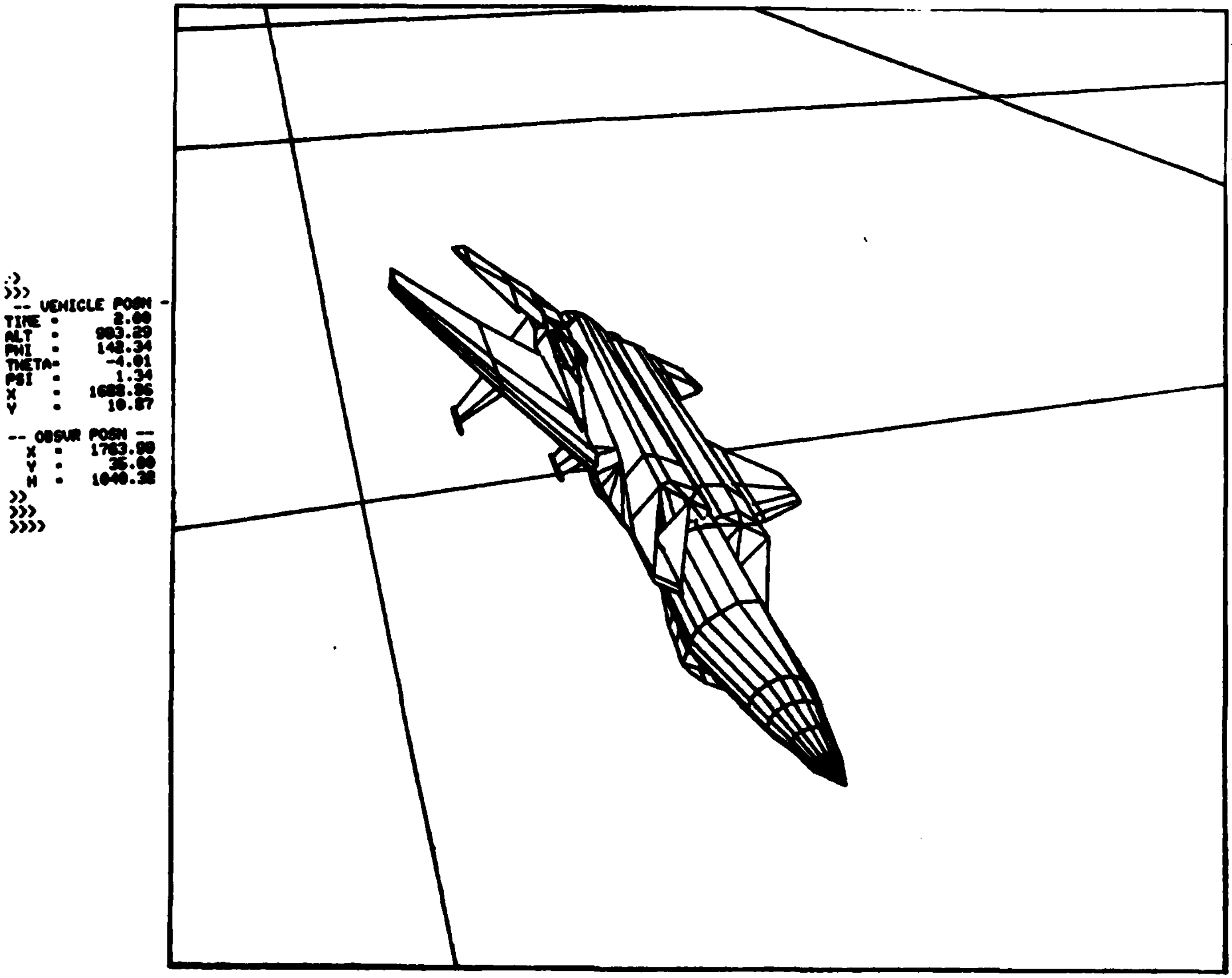


FIGURE 16
CHASE PLANE MODE - T=2

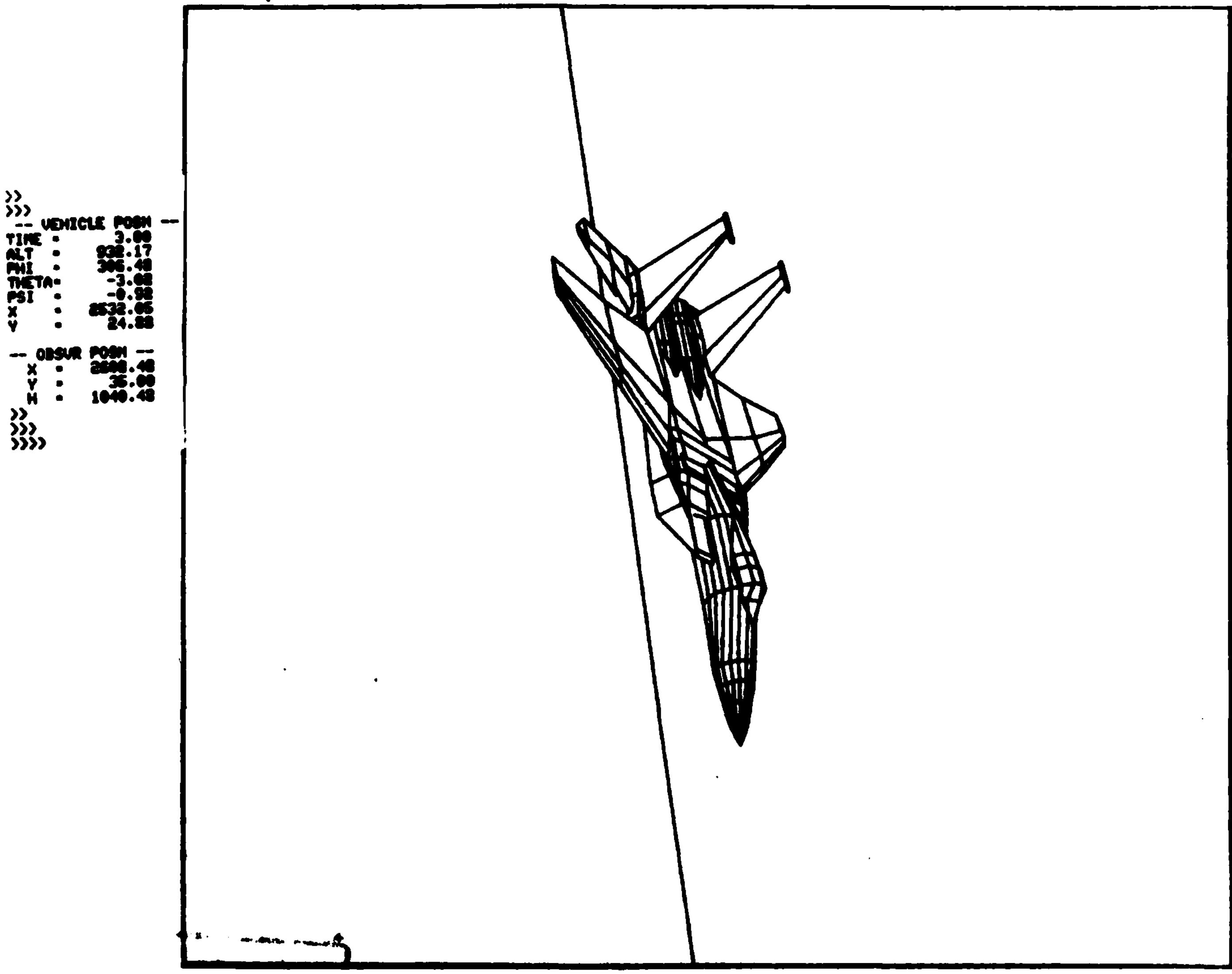


FIGURE 17
CHASE PLANE MODE - T=3

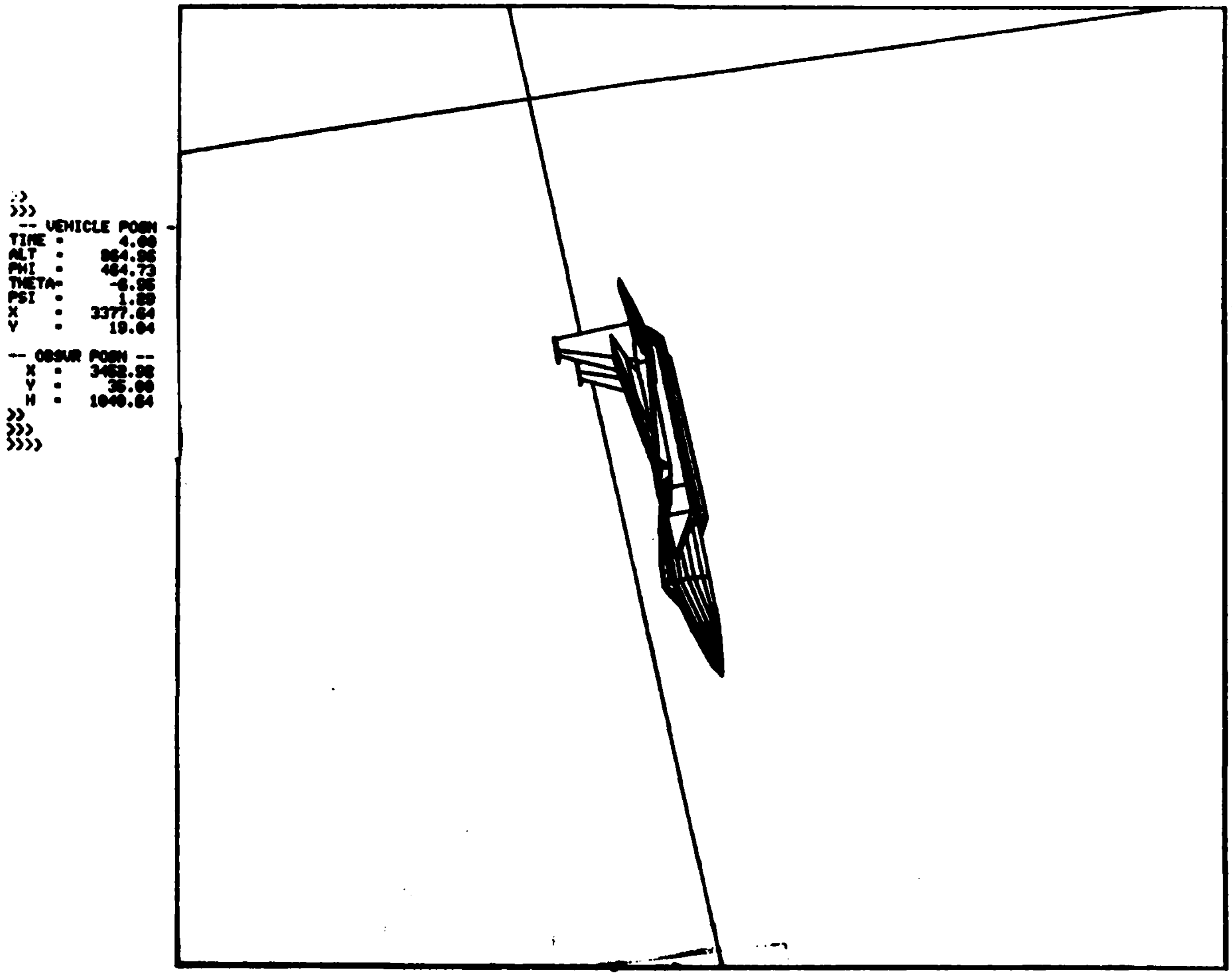


FIGURE 18
CHASE PLANE MODE - T=4

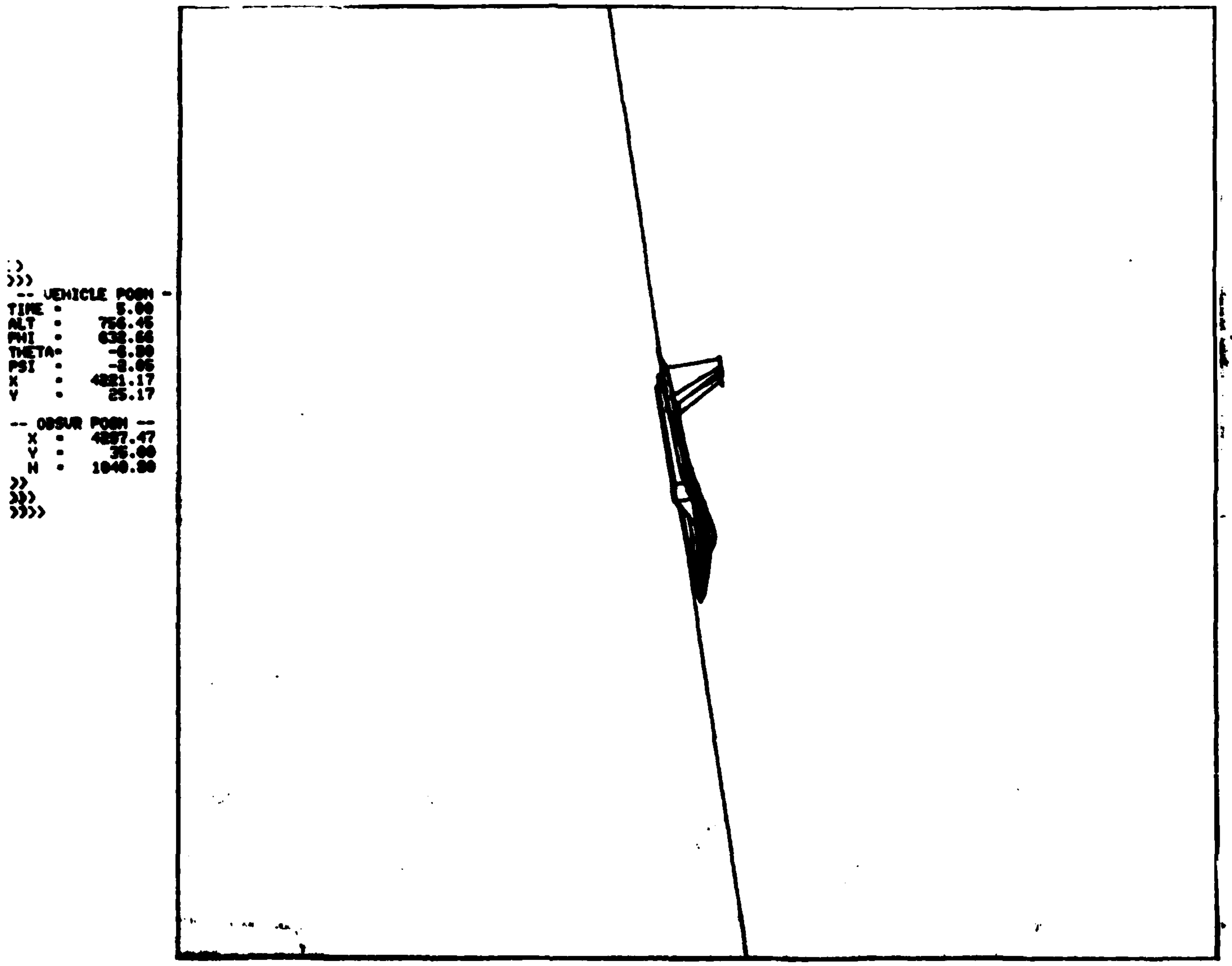


FIGURE 19
CHASE PLANE MODE - T=5

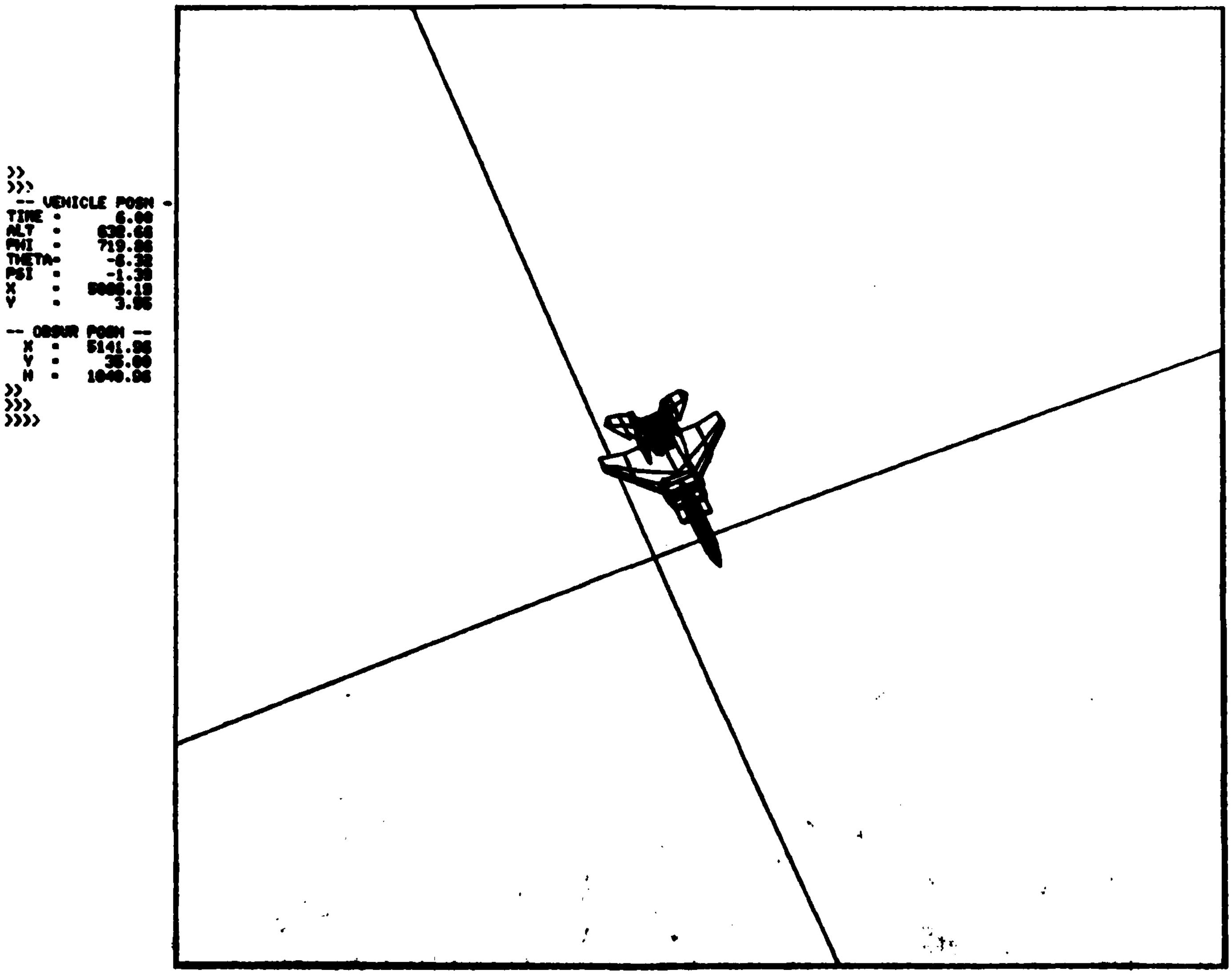


FIGURE 20
CHASE PLANE MODE - T=6

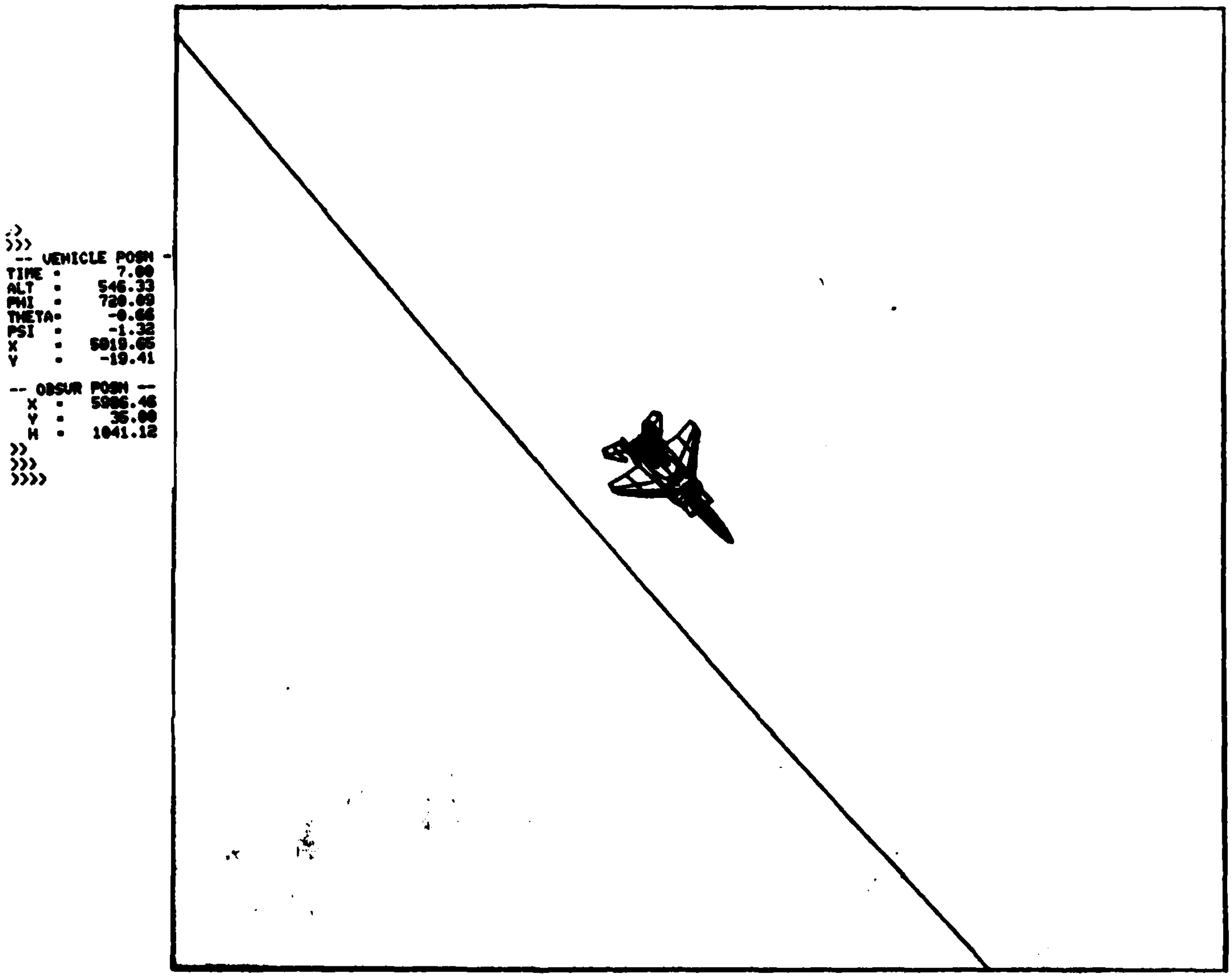


FIGURE 21
CHASE PLANE MODE - T=7

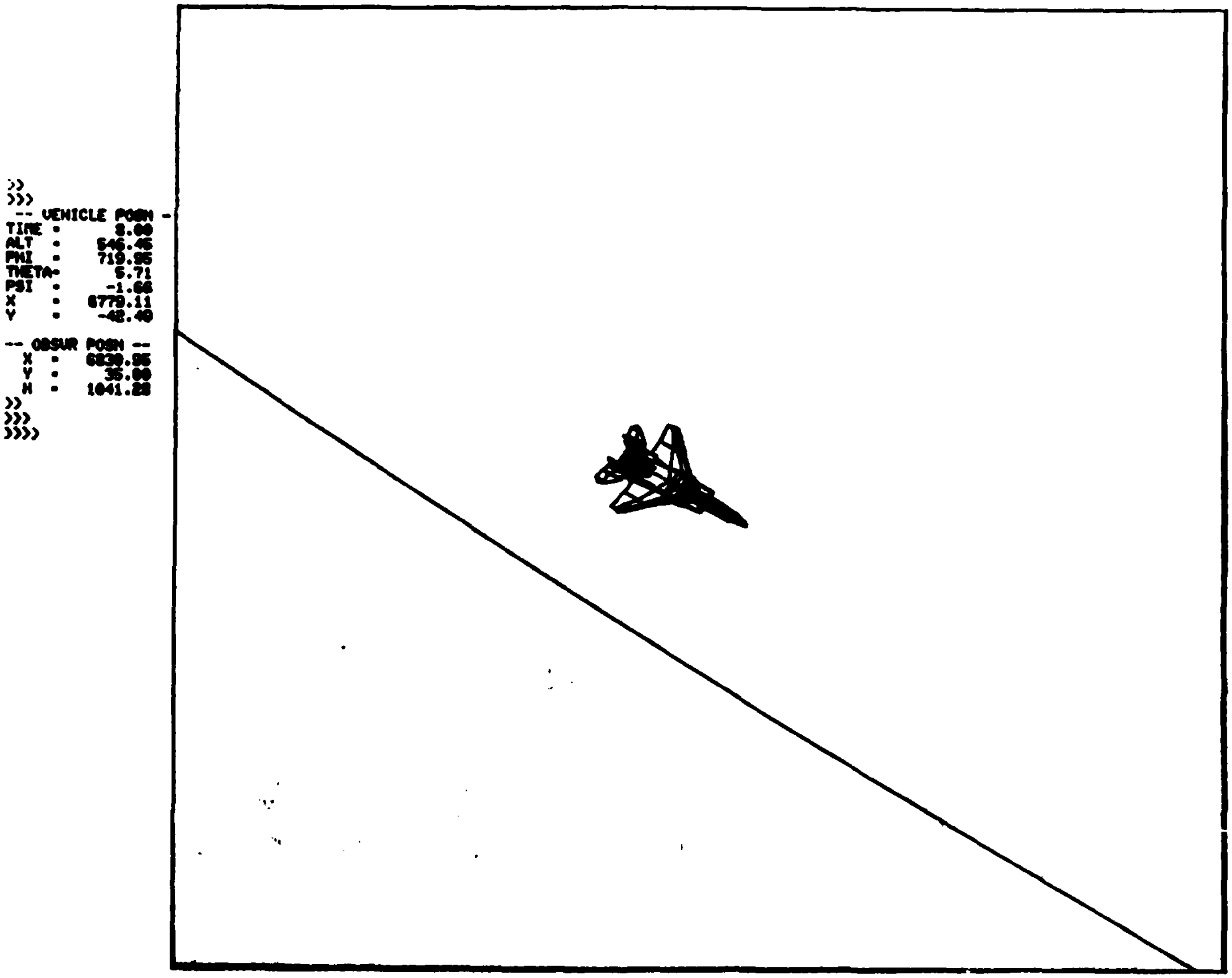


FIGURE 22
CHASE PLANE MODE - T=8

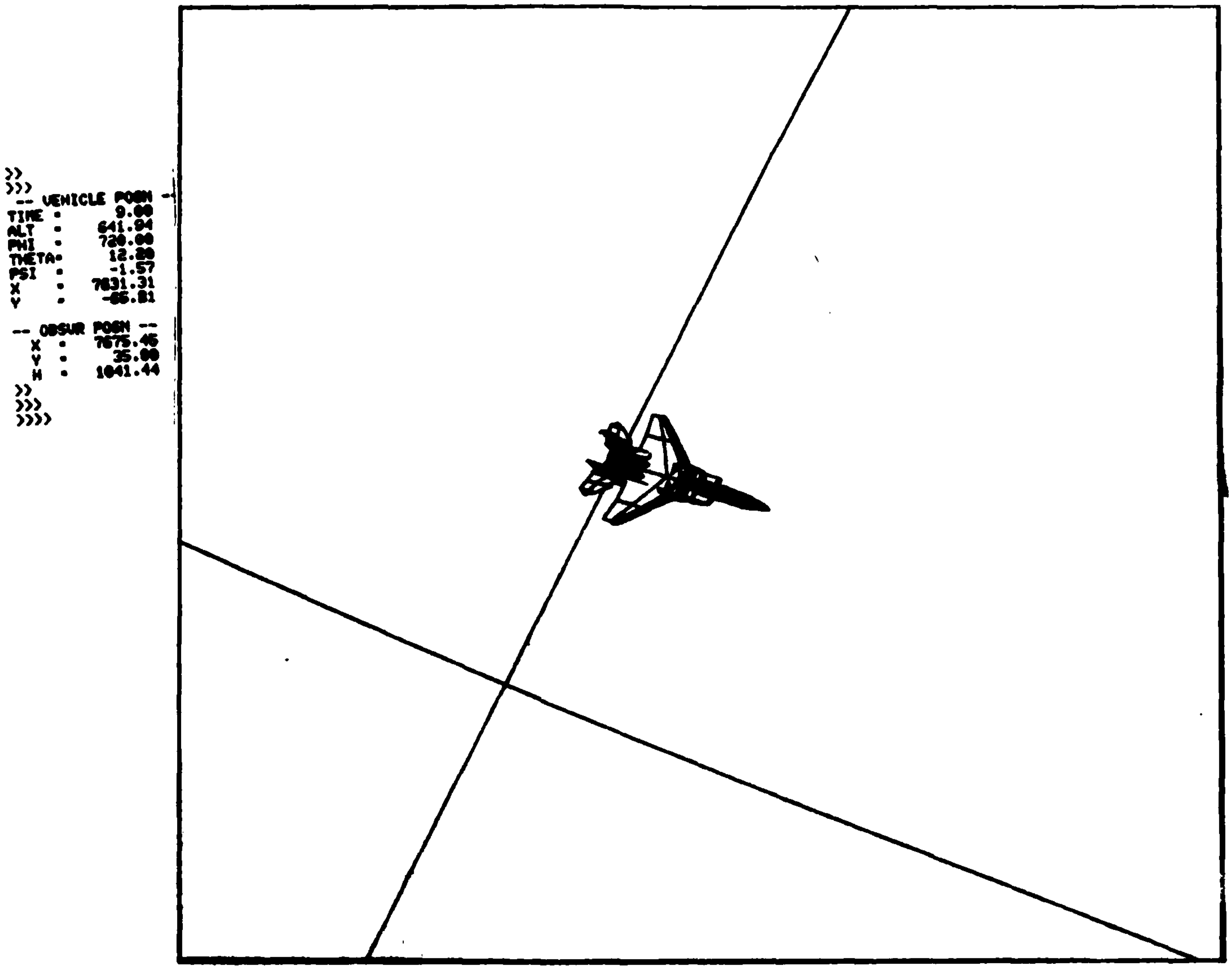


FIGURE 23
CHASE PLANE MODE - T=9

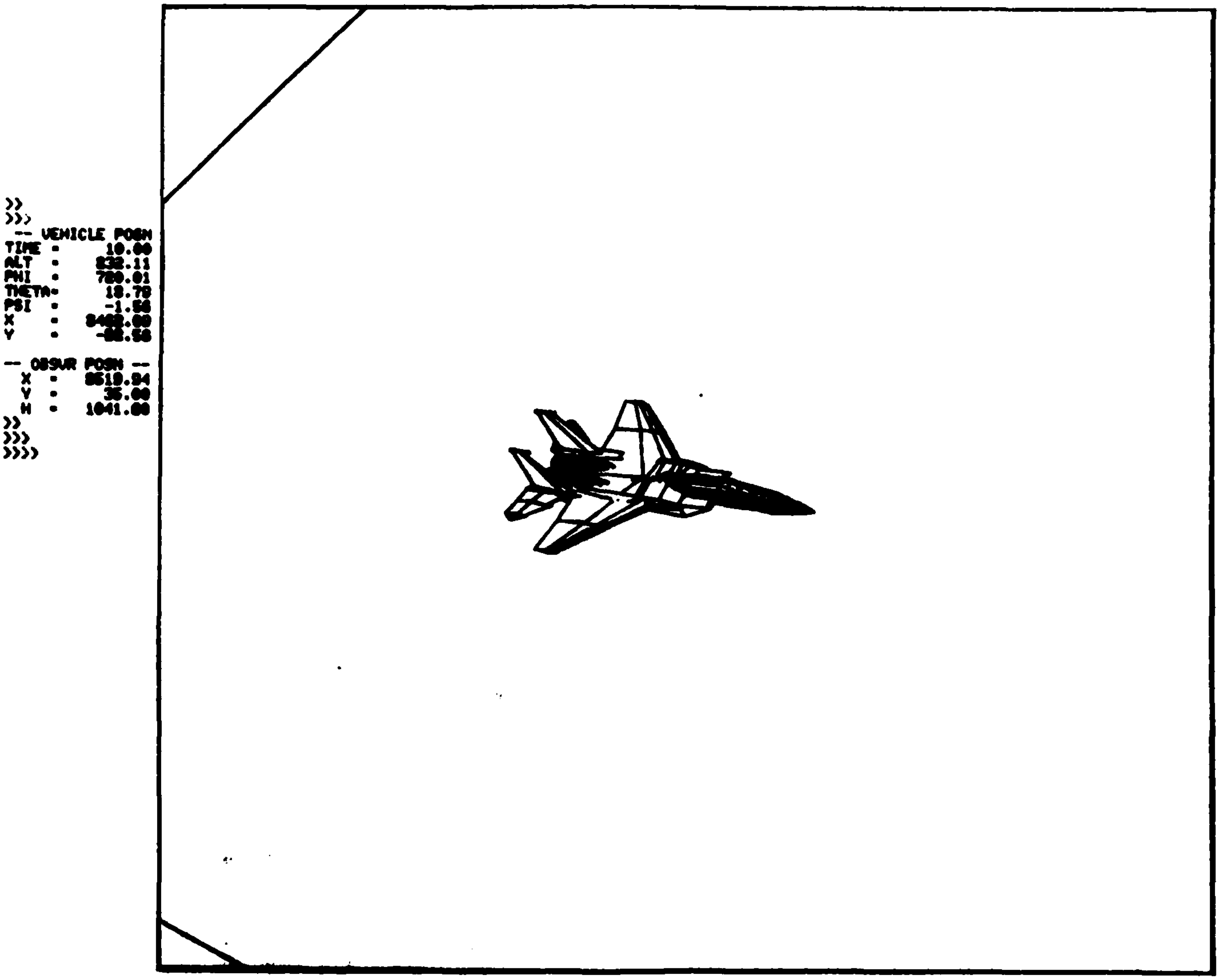


FIGURE 24
CHASE PLANE MODE - T=10

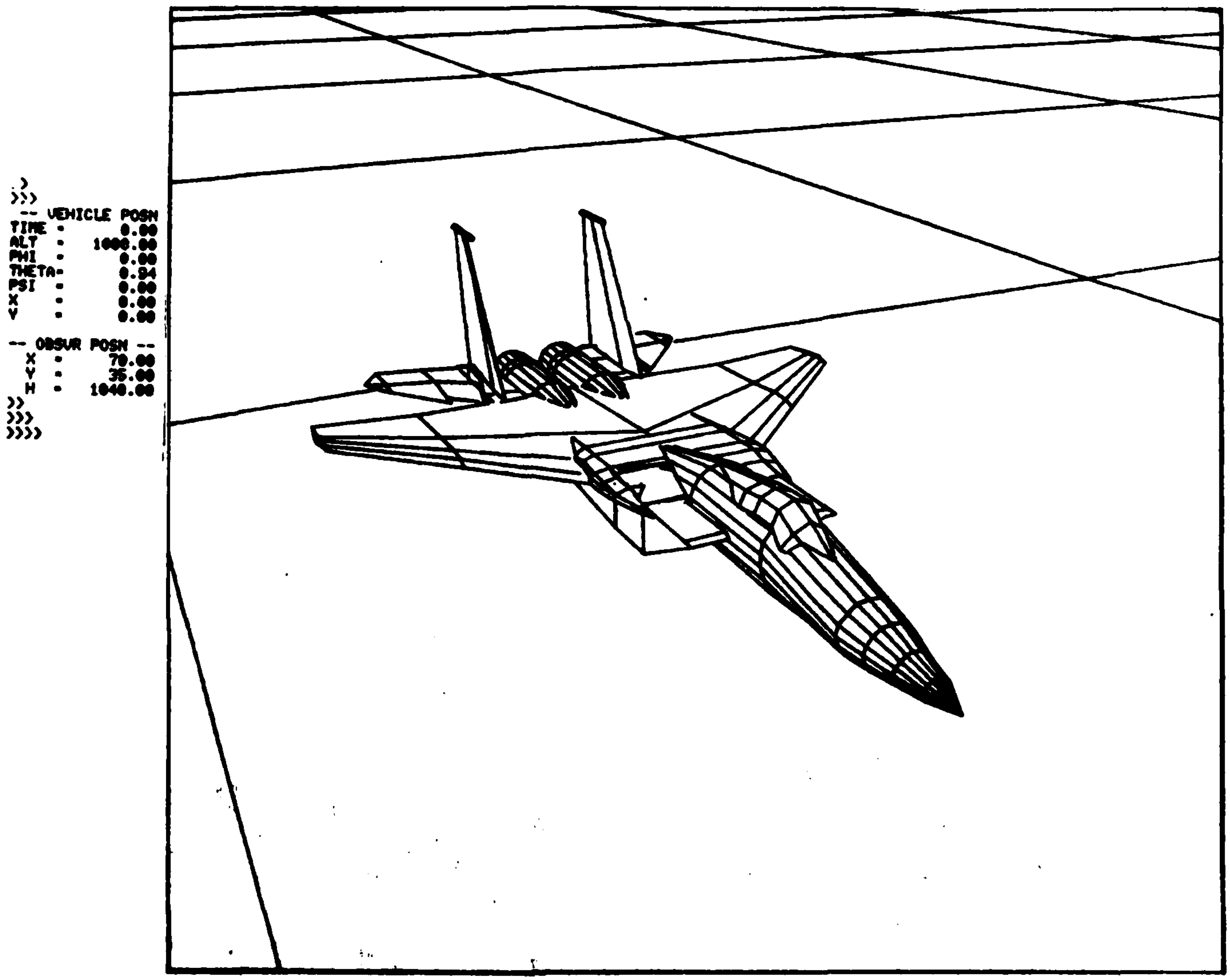


FIGURE 25
WINGMAN MODE - T=0

```
>>
>>>
 -- VEHICLE POSN
TIME  =      1.00
ALT   =   1000.16
PHI   =      0.00
THETA=       0.96
PSI   =      0.00
X     =    844.49
Y     =      0.00

-- OBSVR POSN --
  X   =    914.49
  Y   =     35.00
  H   =   1040.16
>>
>>>
>>>>
```

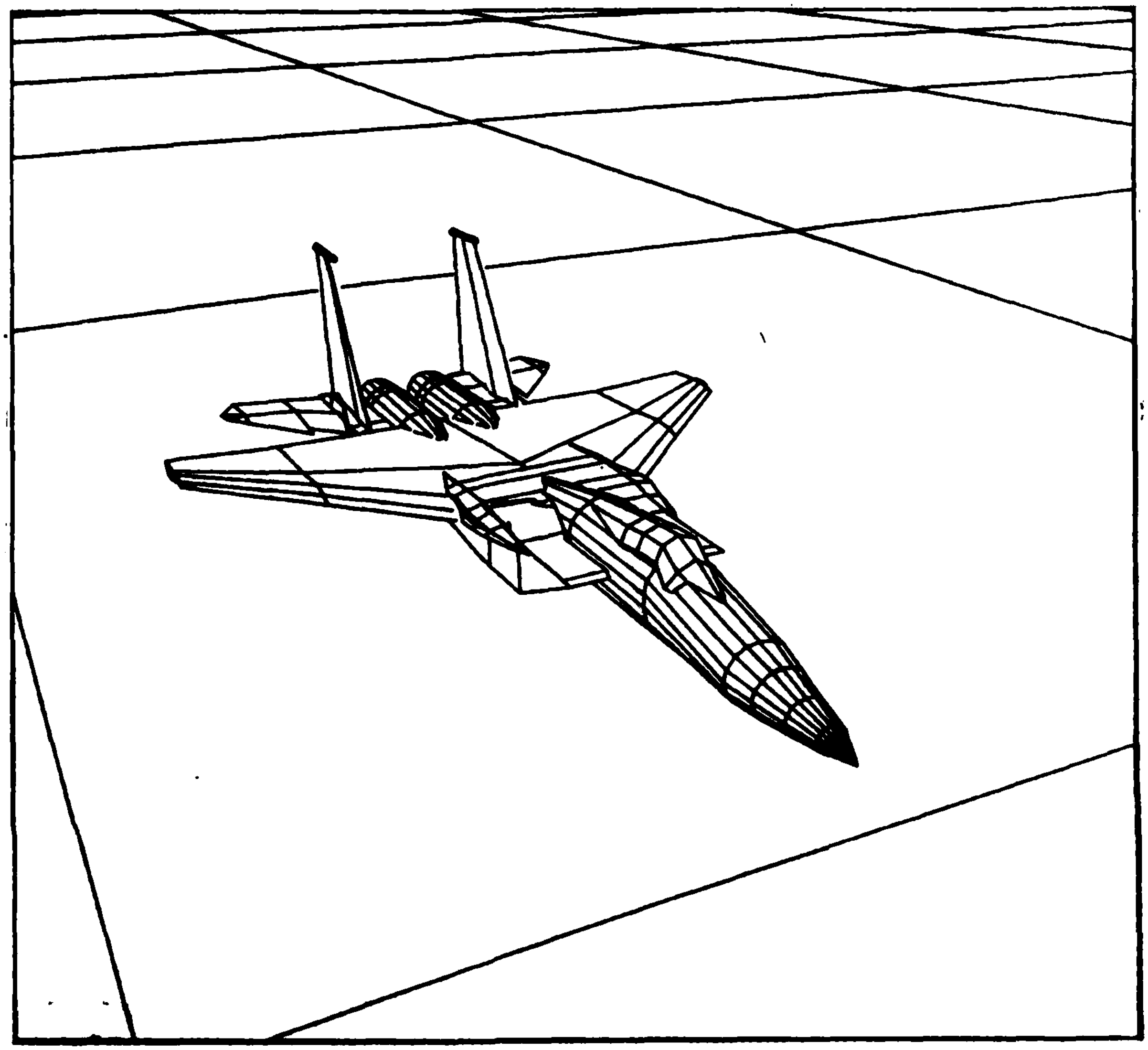


FIGURE 26
WINGMAN MODE - T=1

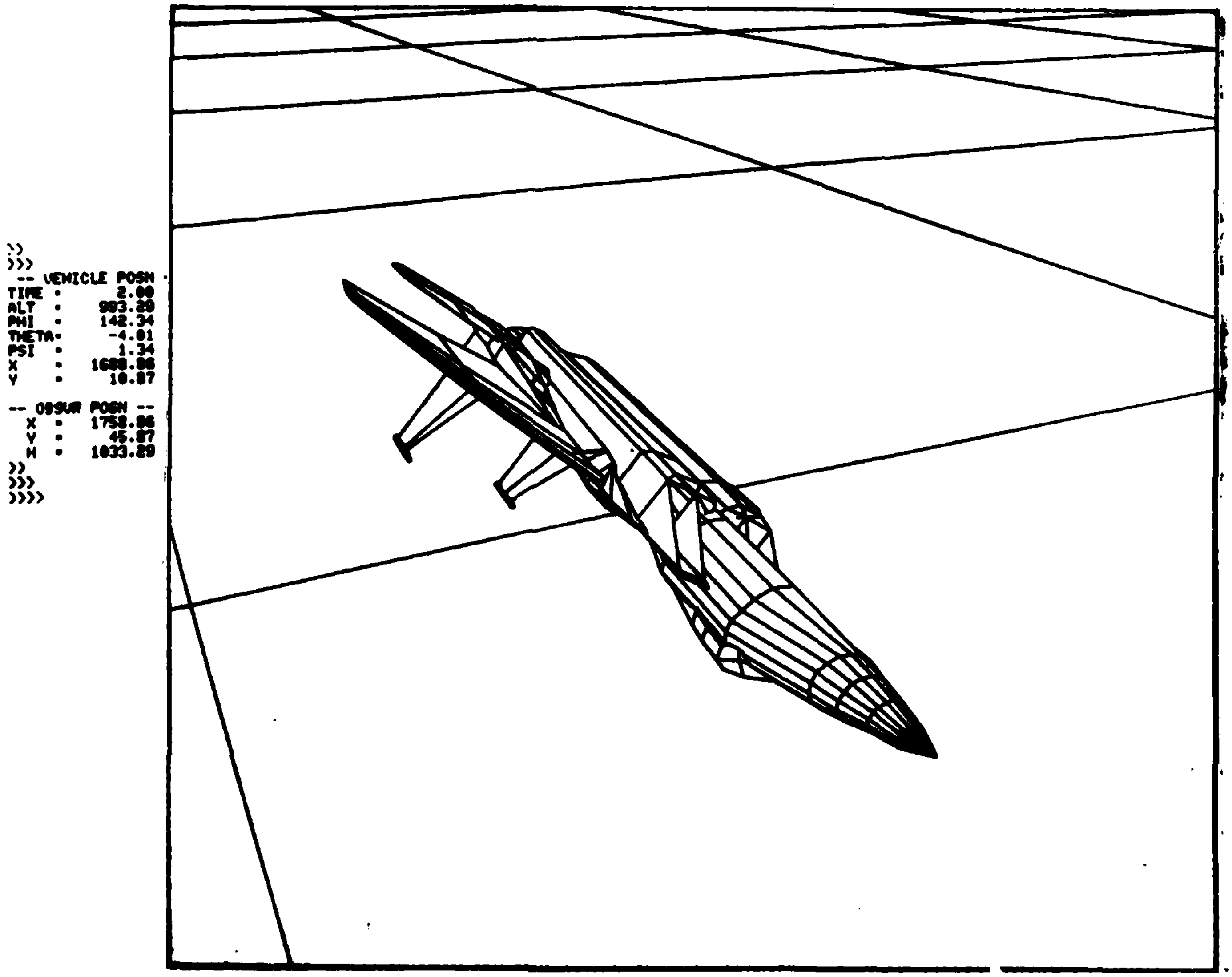


FIGURE 27
WINGMAN MODE - T=2

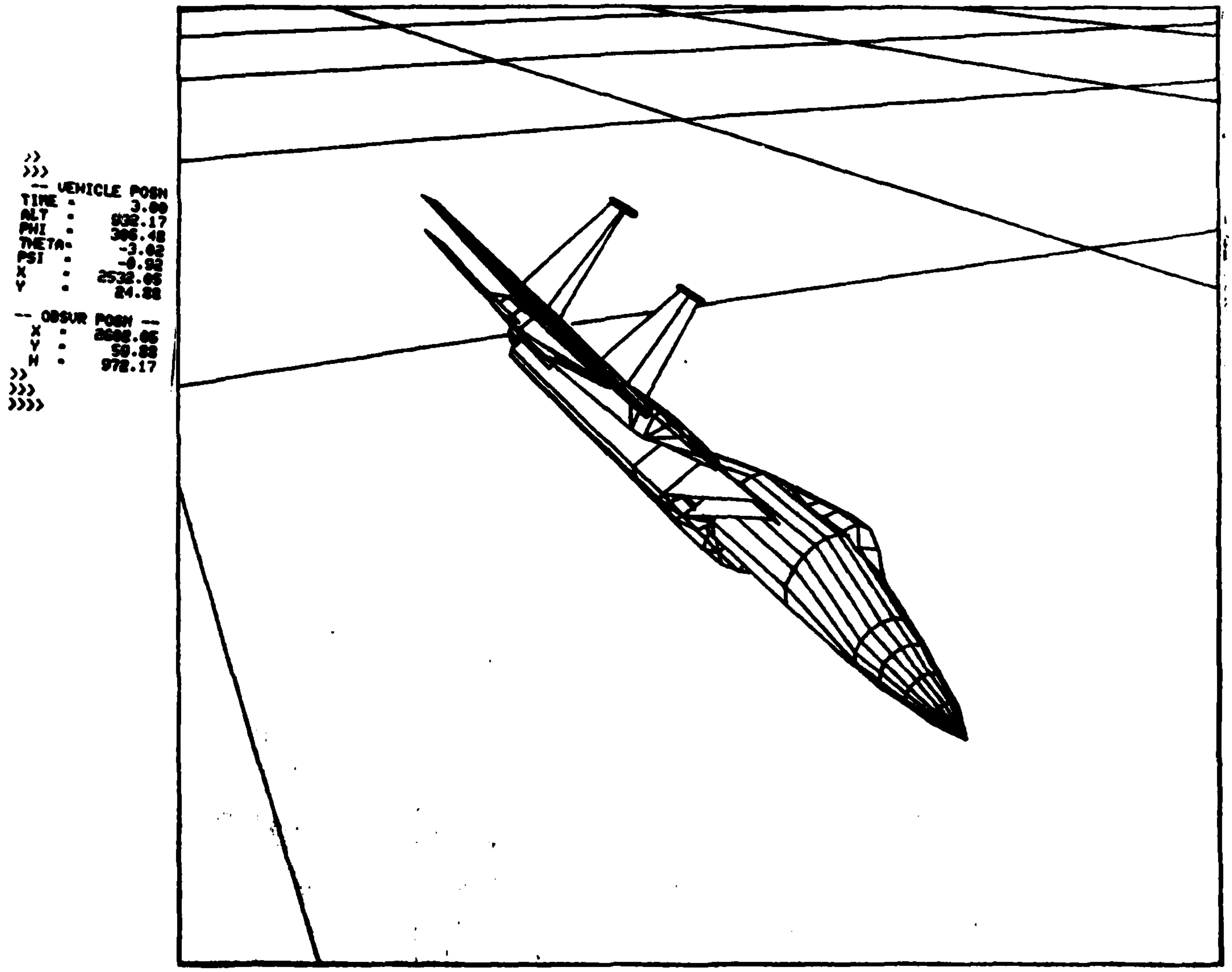


FIGURE 28
WINGMAN MODE - T=3

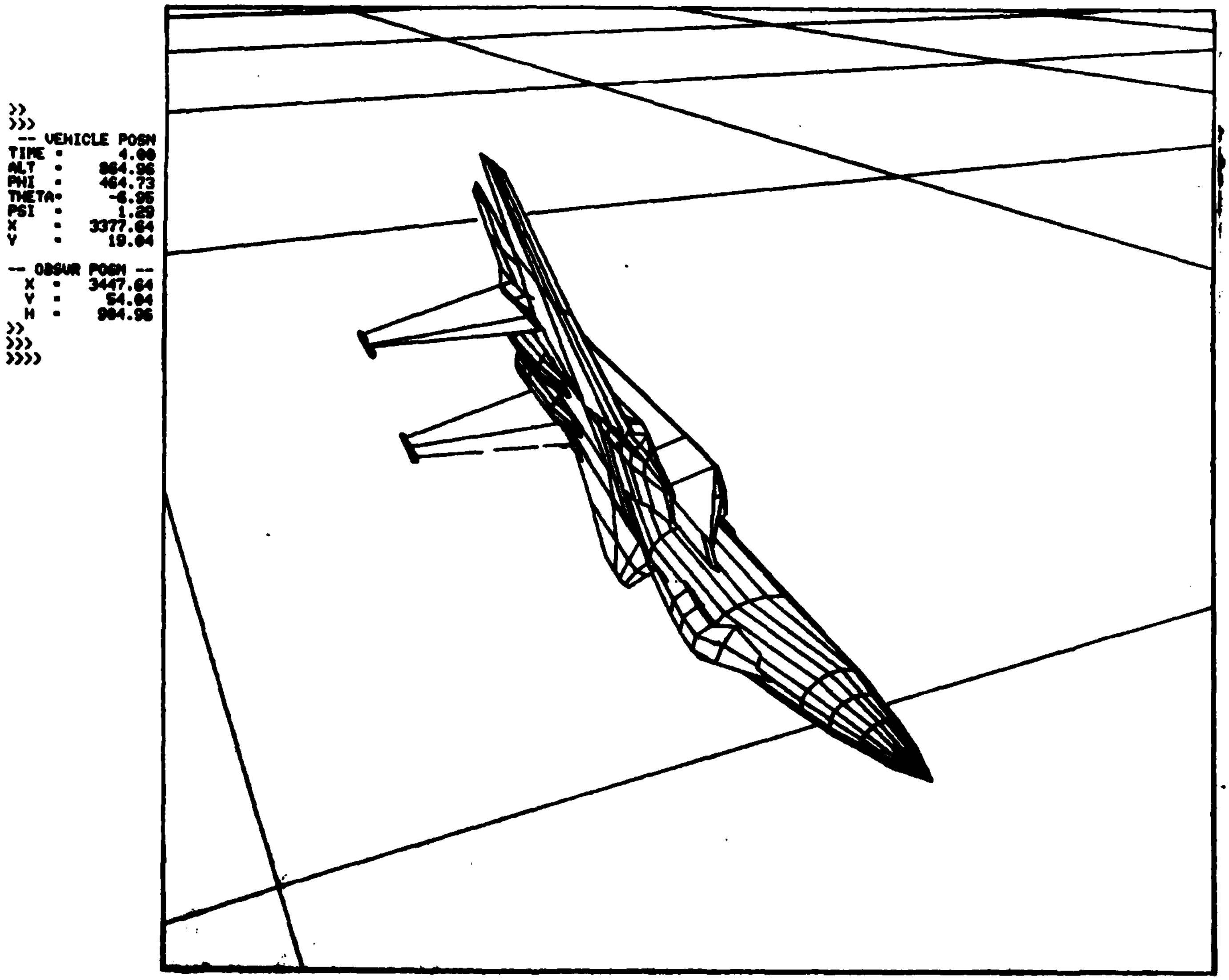


FIGURE 29
WINGMAN MODE - T=4

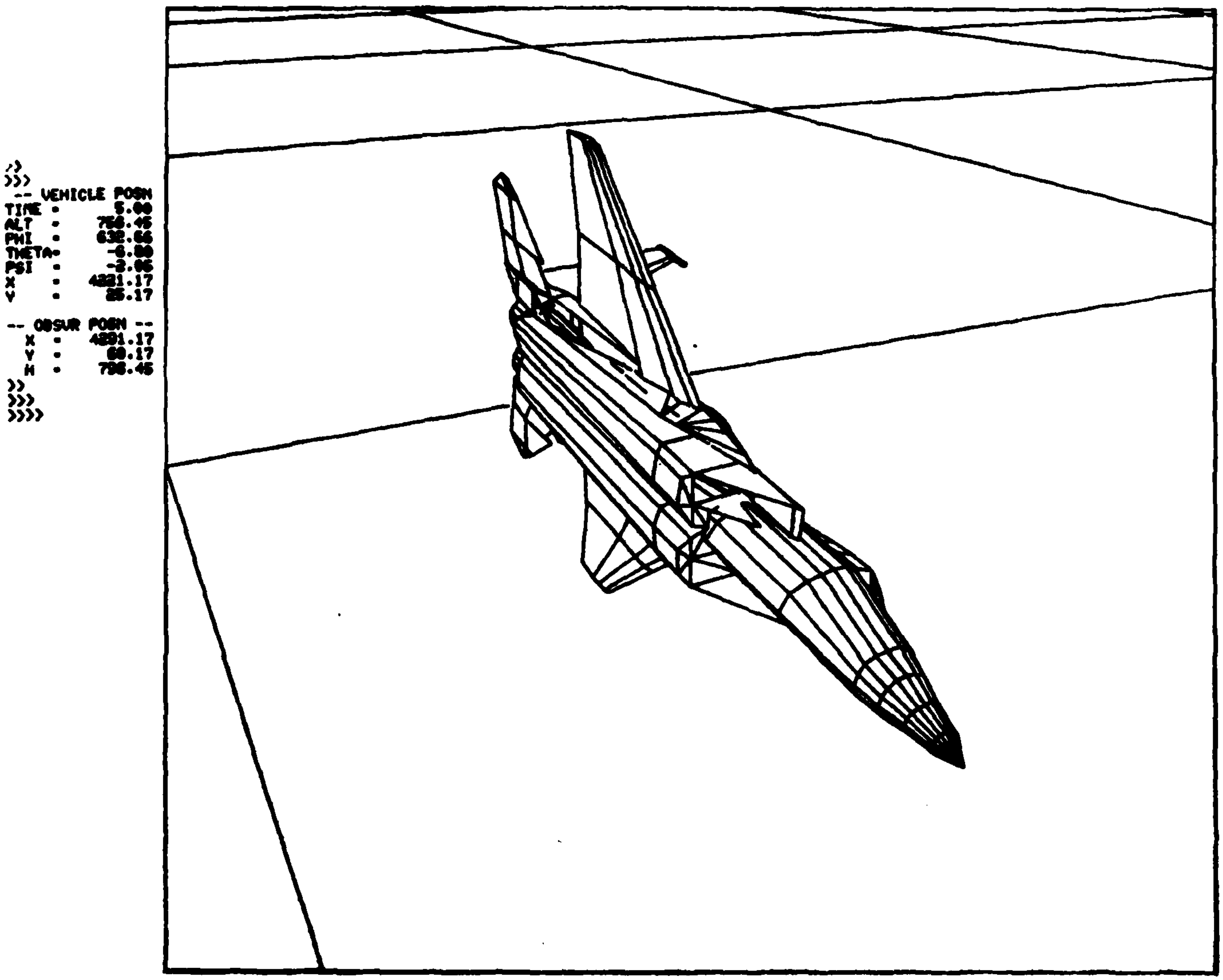


FIGURE 30
WINGMAN MODE - T=5

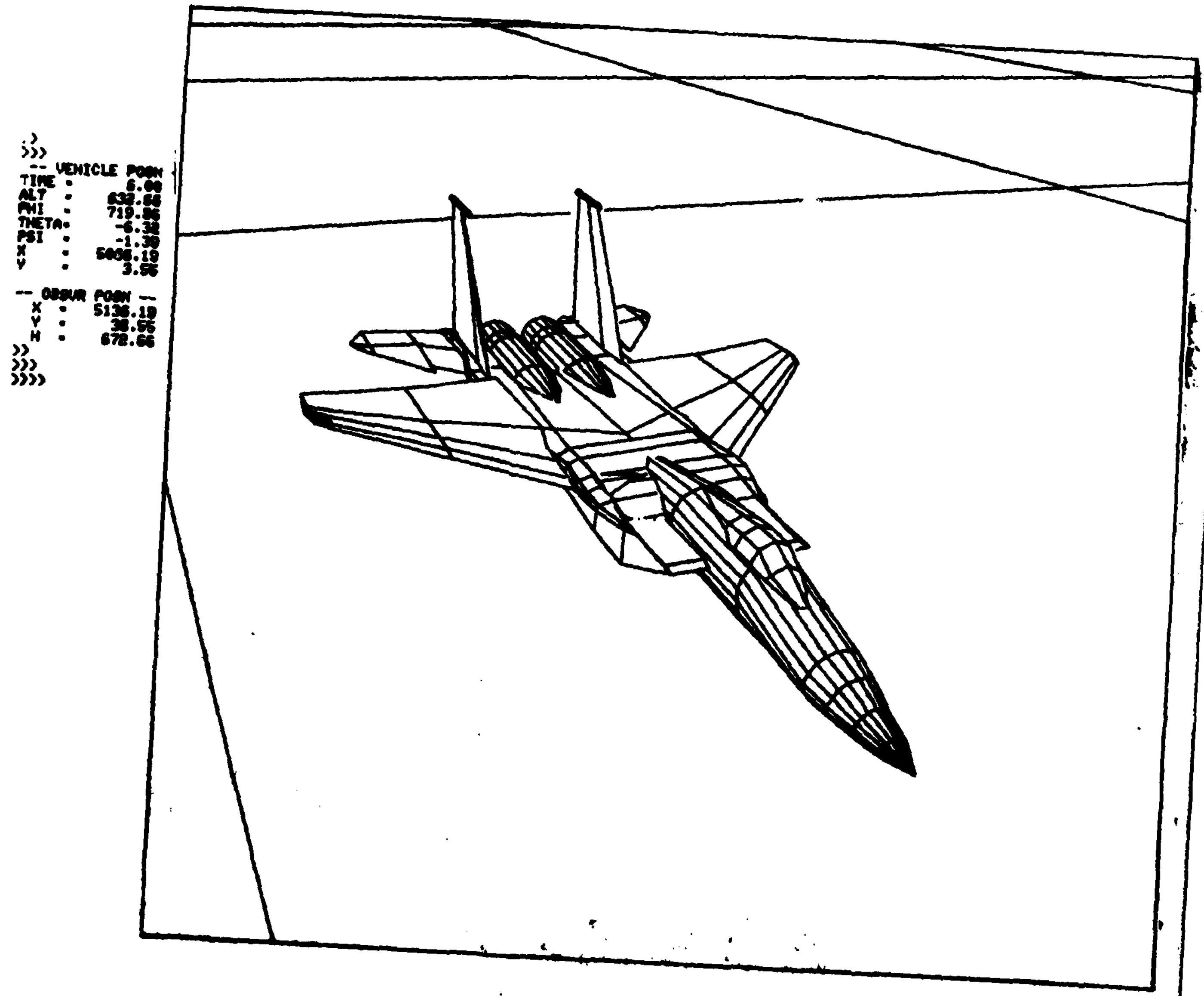


FIGURE 31
WINGMAN MODE - T=6

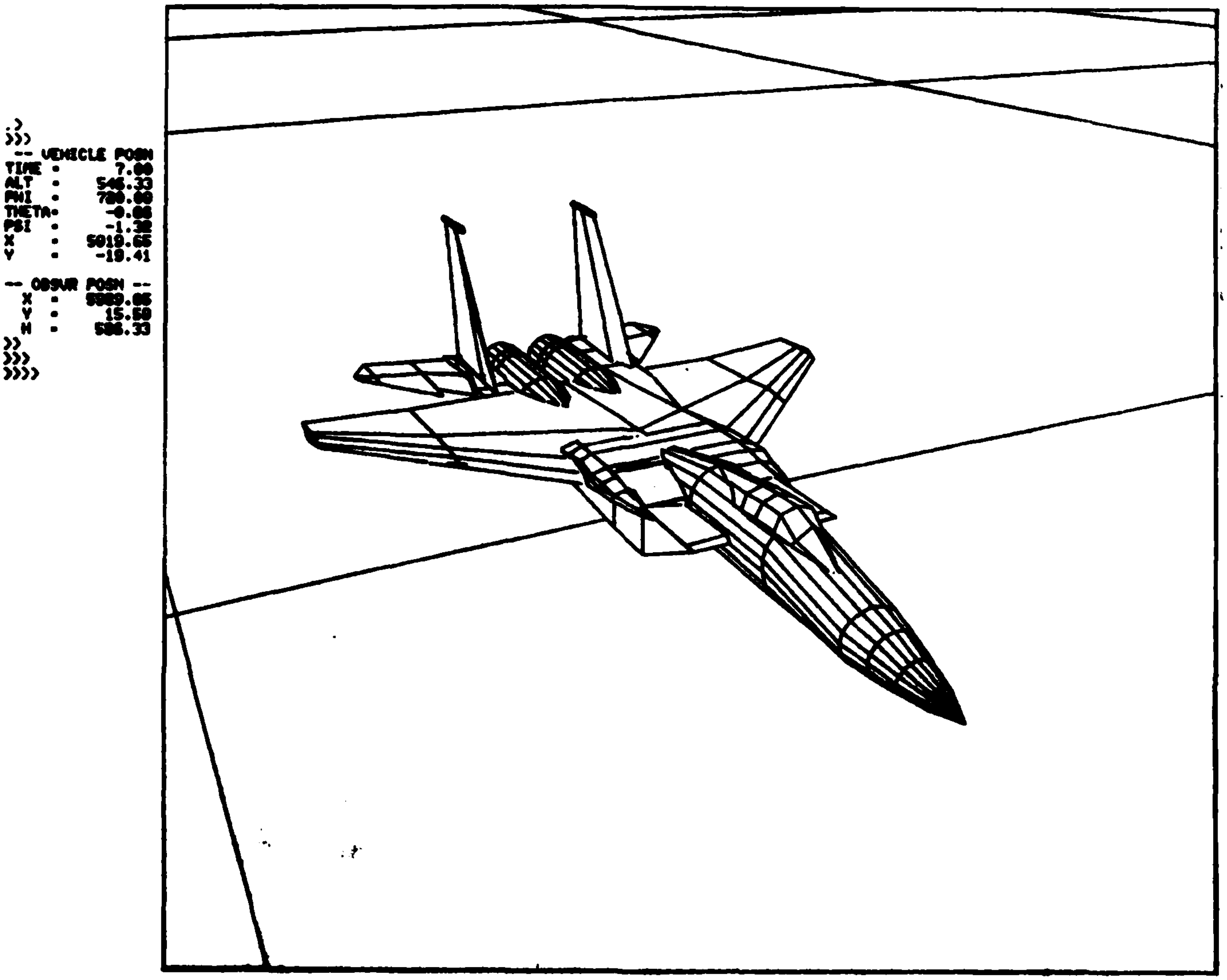


FIGURE 32
WINGMAN MODE - T=7

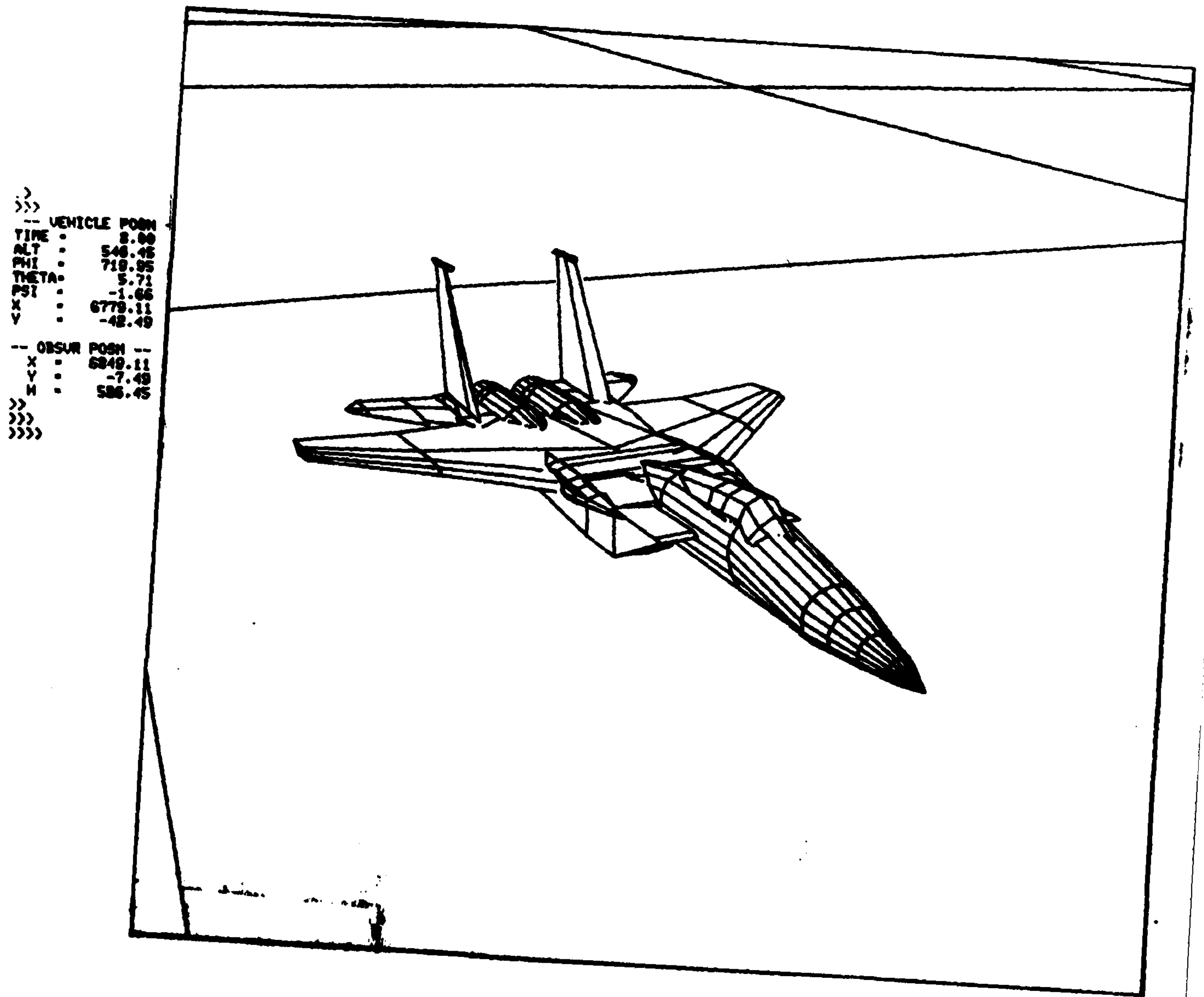


FIGURE 33
WINGMAN MODE - T=8

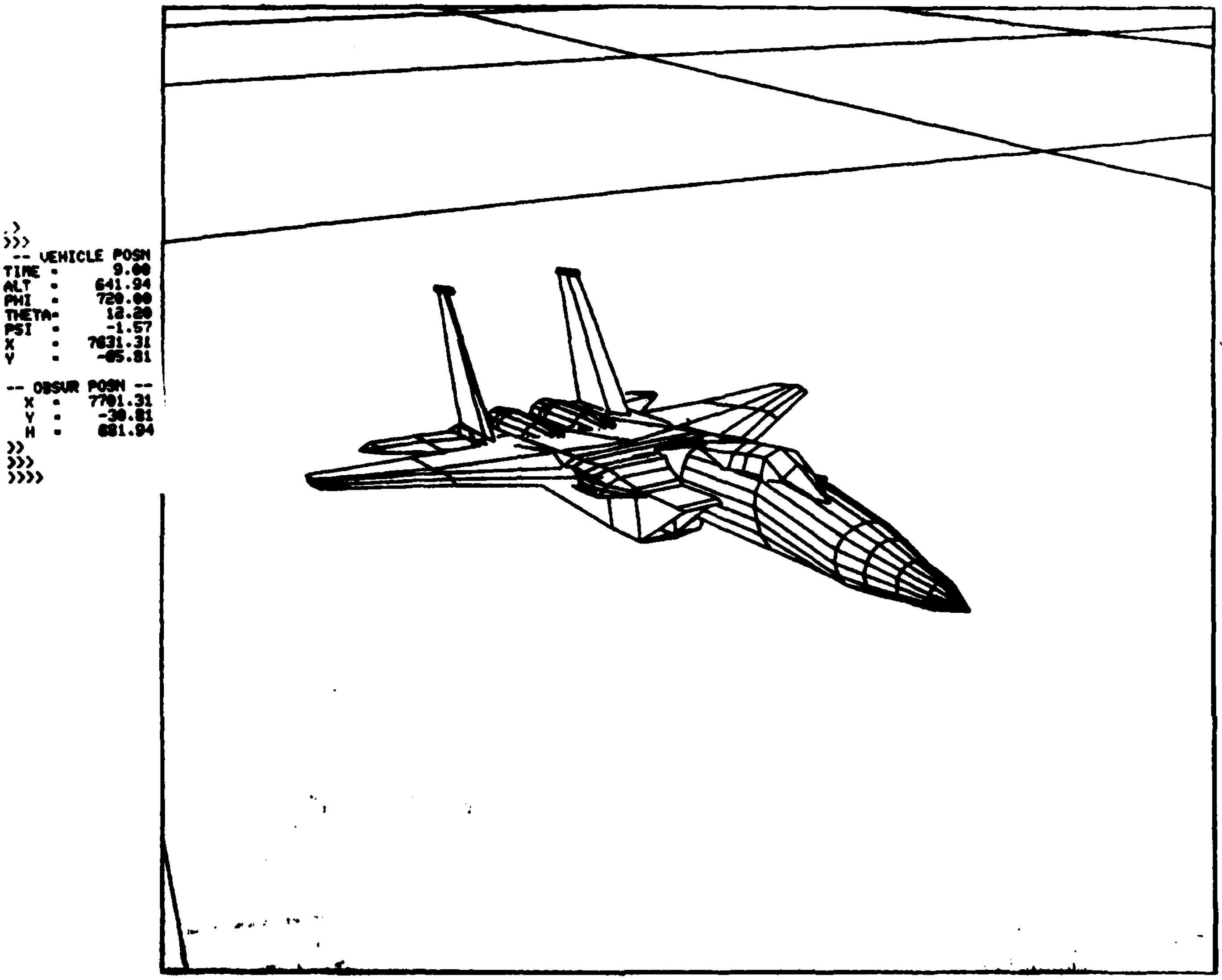


FIGURE 34
WINGMAN MODE - T=9

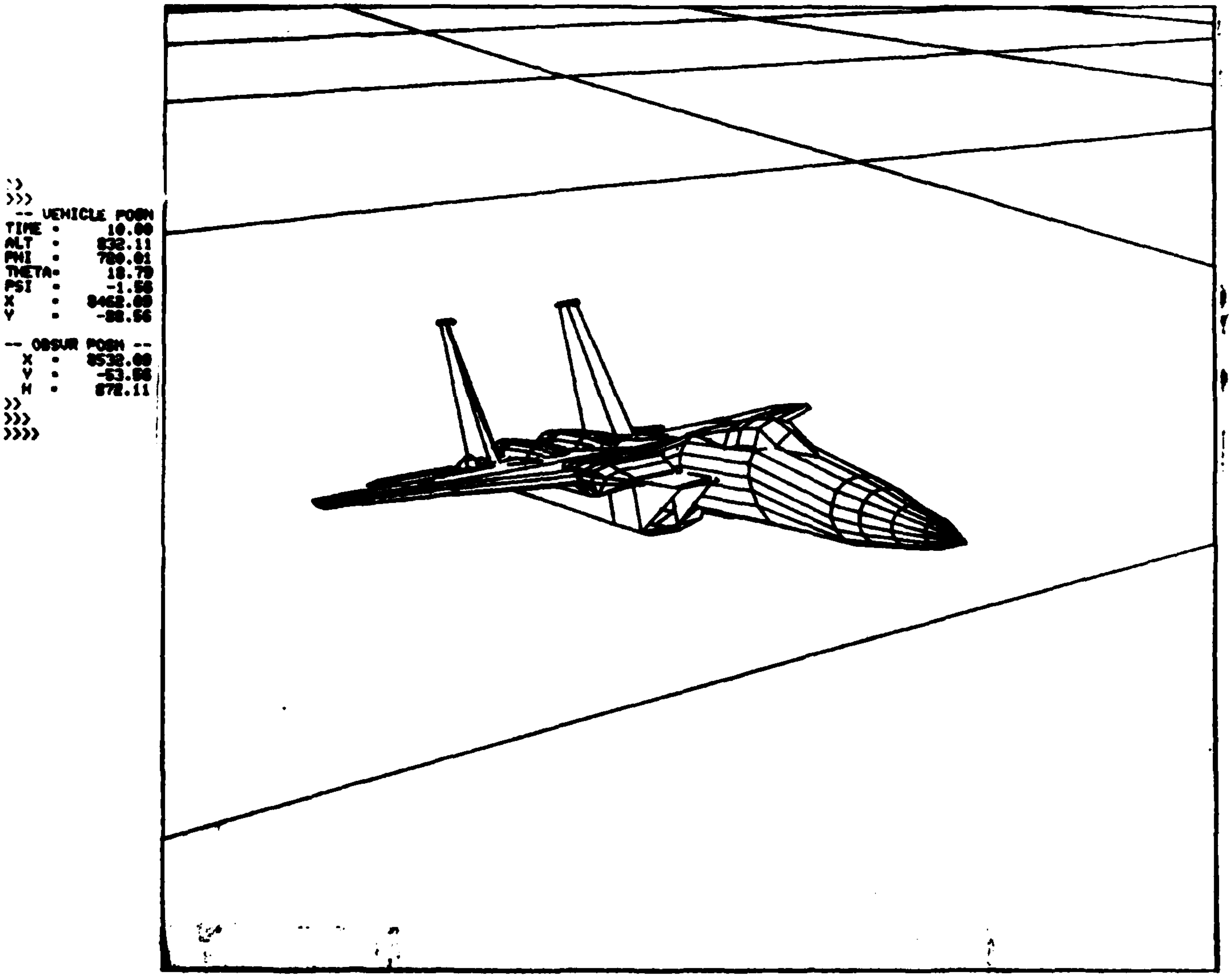


FIGURE 35
WINGMAN MODE - T=10

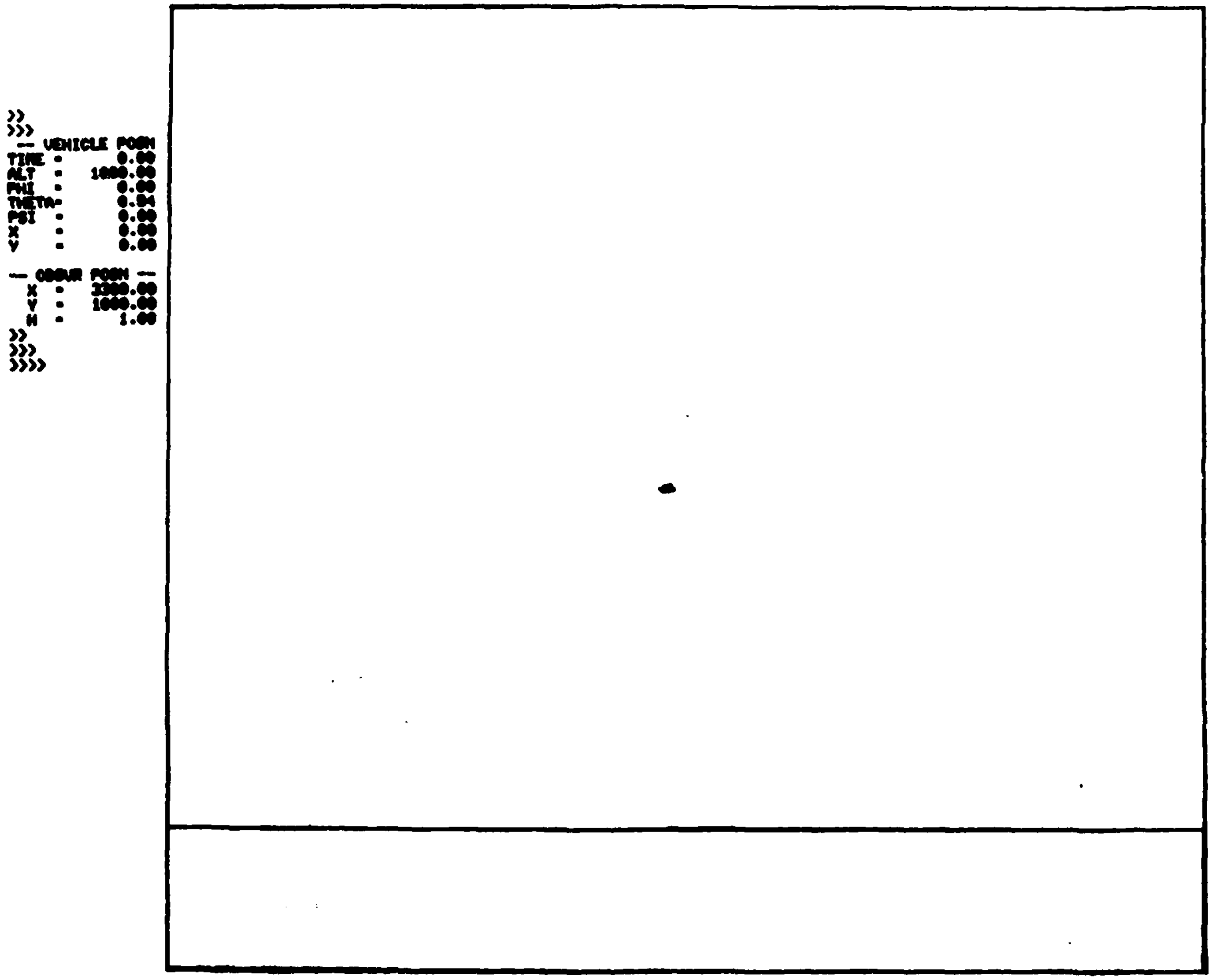
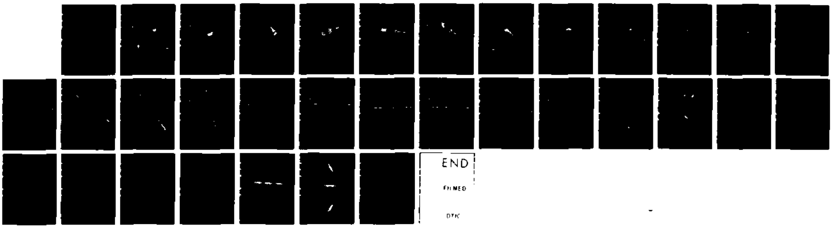
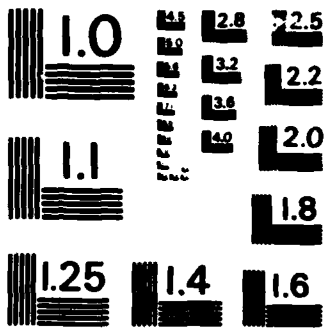


FIGURE 36
FIXED POSITION MODE - T=0 (ZOOM=1)

AD-A146 525   COMPUTER ANIMATED REPRESENTATIONS TO OPTICALLY OBSERVE NUMERICAL EVALUATI..(U) AERONAUTICAL SYSTEMS DIV WRIGHT-PATTERSON AFB OH DIRECTORATE O..   2/2

UNCLASSIFIED   M J MIEDLAR ET AL. APR 83   F/G 9/2   NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

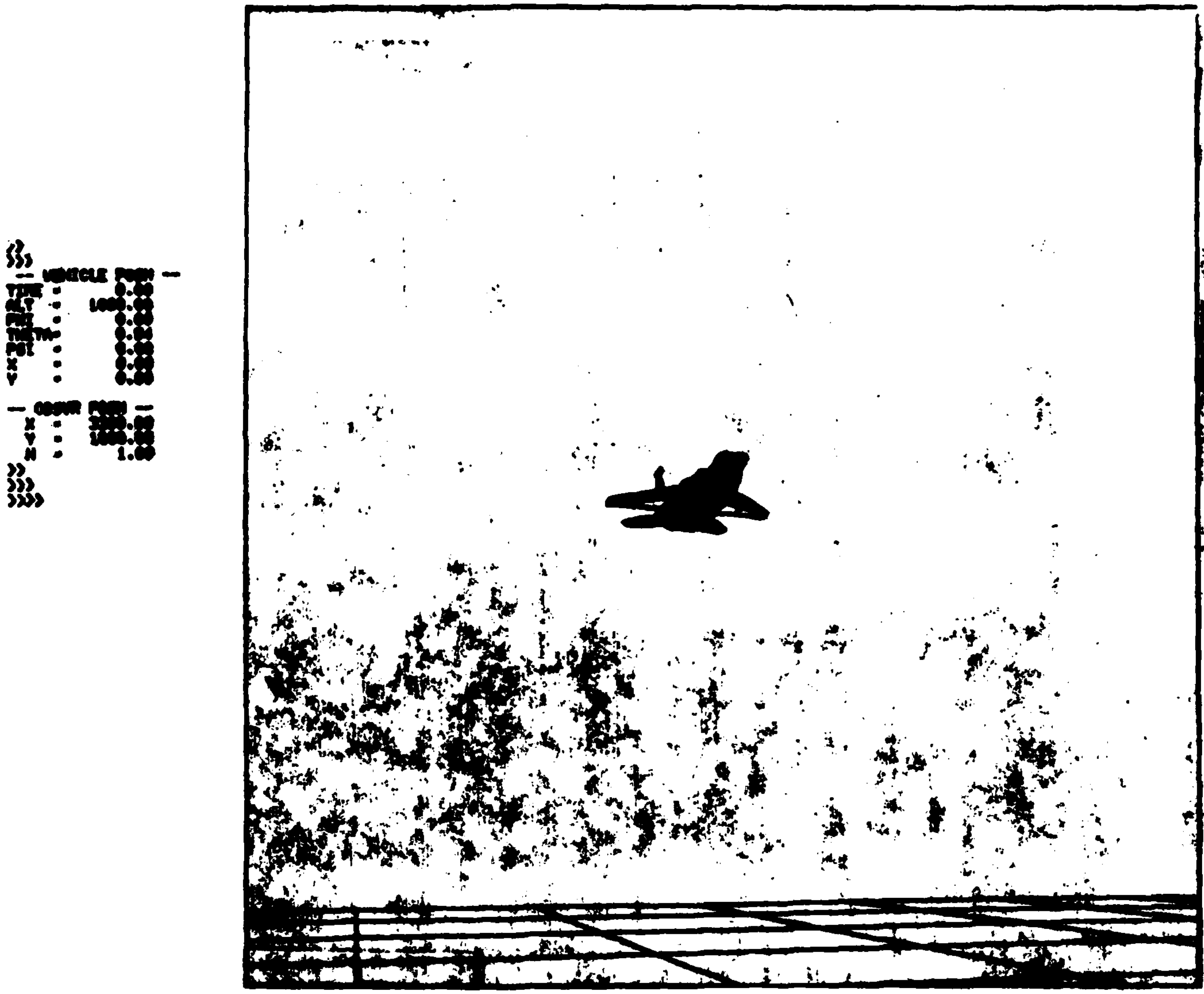


FIGURE 37
FIXED POSITION MODE - T=0 (ZOOM=12.5)

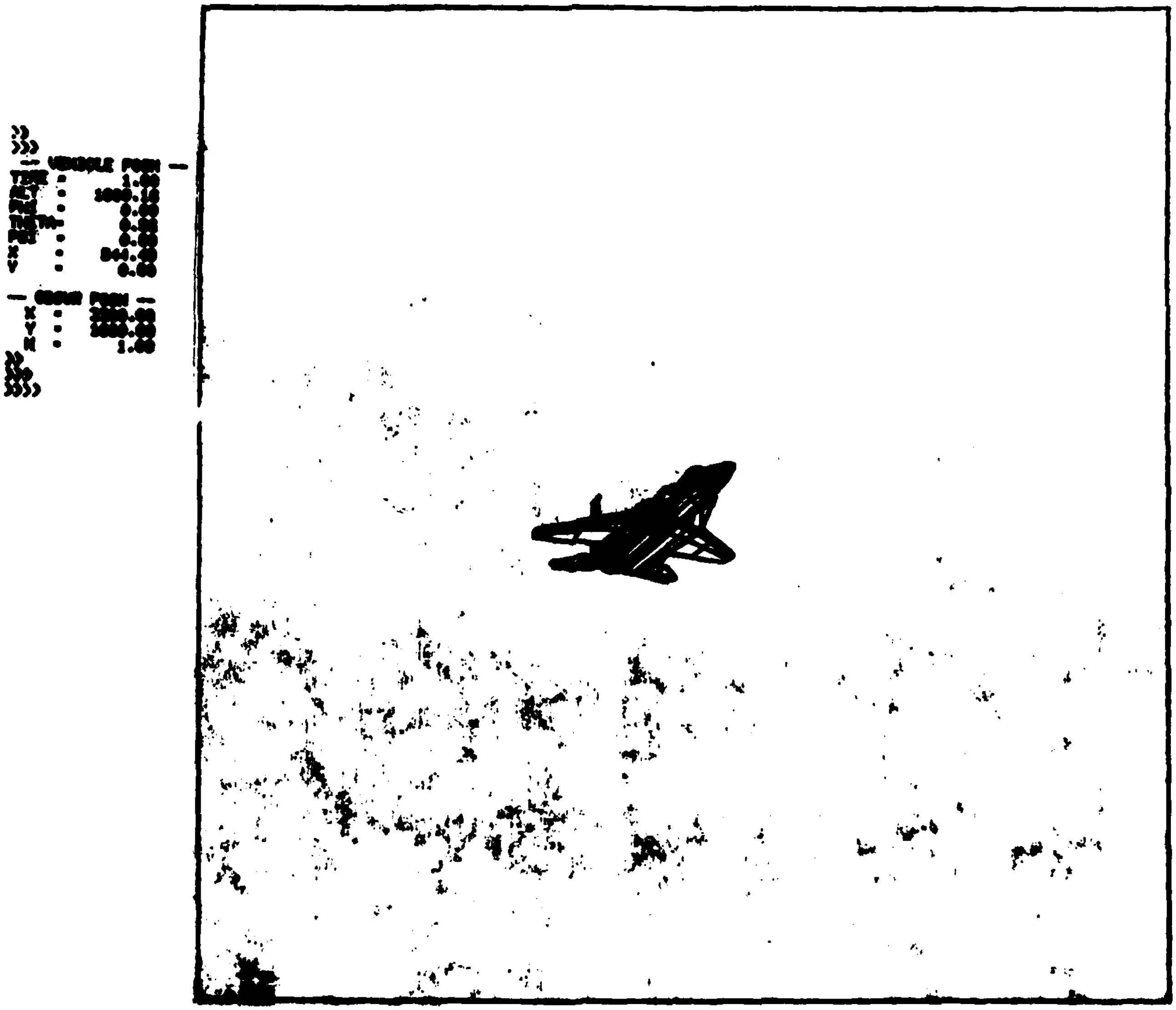


FIGURE 38
FIXED POSITION MODE - T=1

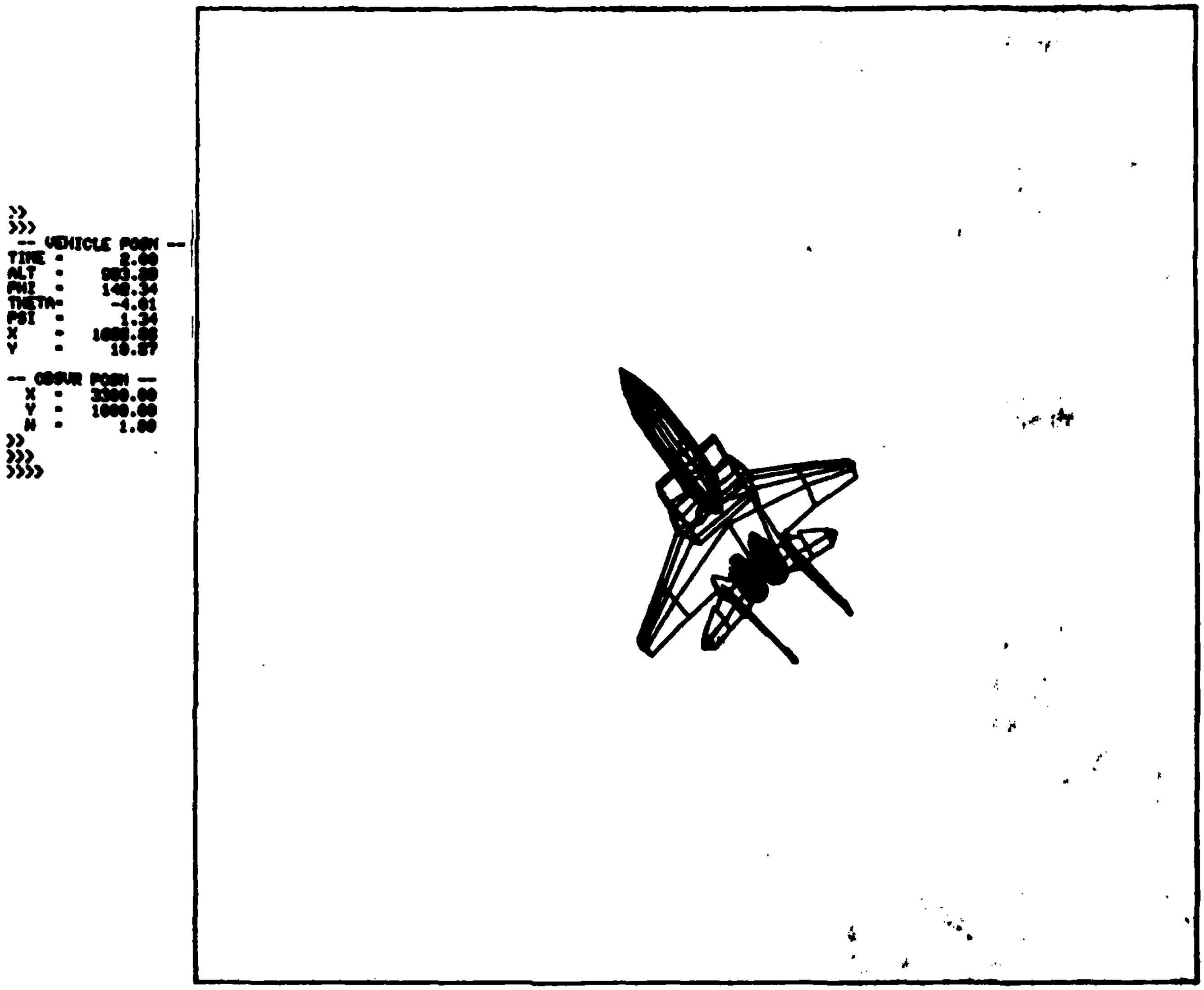


FIGURE 39
FIXED POSITION MODE - T=2

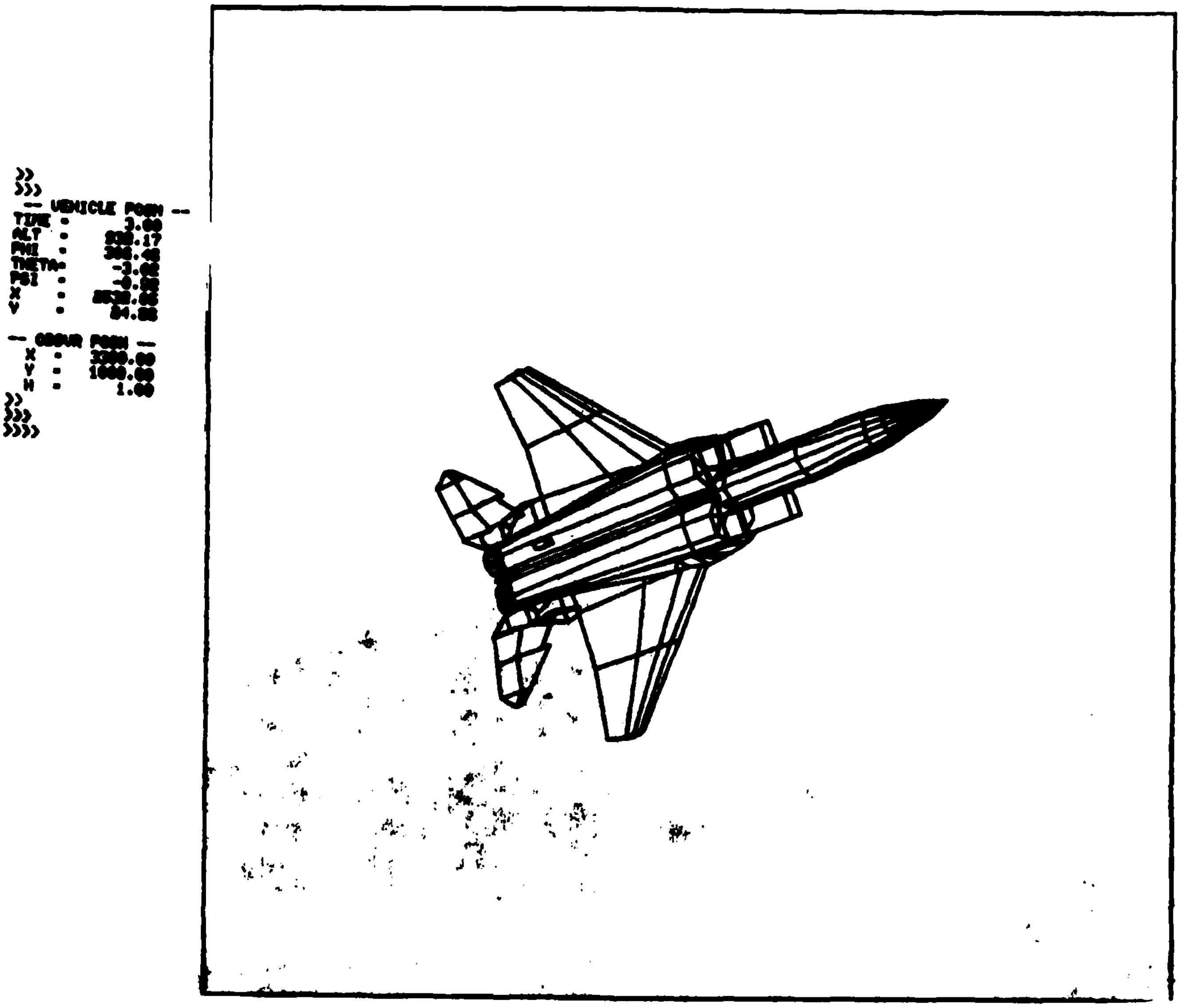


FIGURE 40
FIXED POSITION MODE - T=3

```
>>
>>>
 -- VEHICLE POSN -
TIME  =      4.00
ALT   =    884.86
PHI   =    464.73
THETA =     -6.85
PSI   =      1.00
X     =   3377.04
Y     =     19.04

-- OBSVR POSN --
  X   =   3300.00
  Y   =   1000.00
  H   =      1.00
>>
>>>
>>>>
```

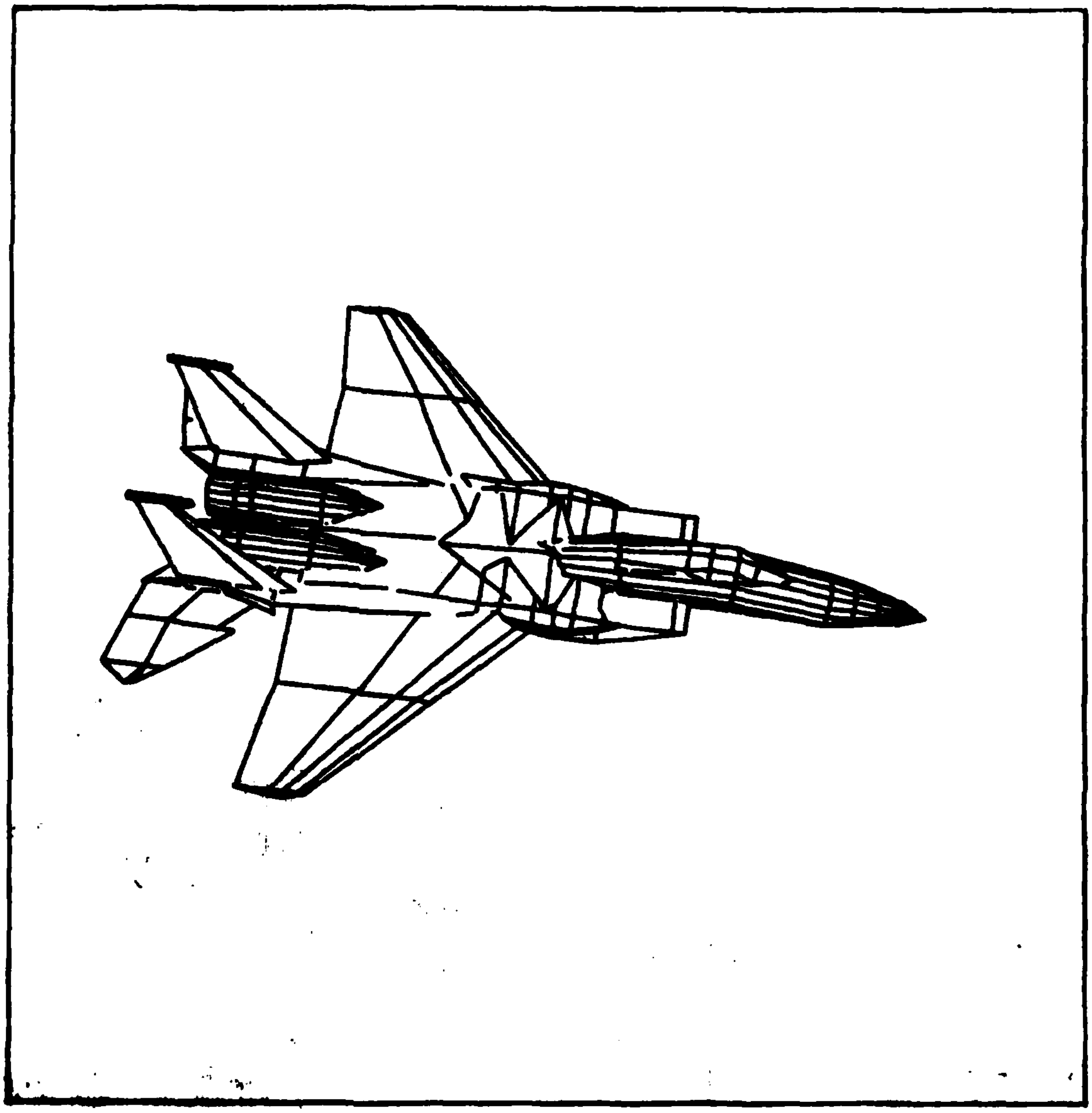


FIGURE 41

FIXED POSITION MODE - T=4

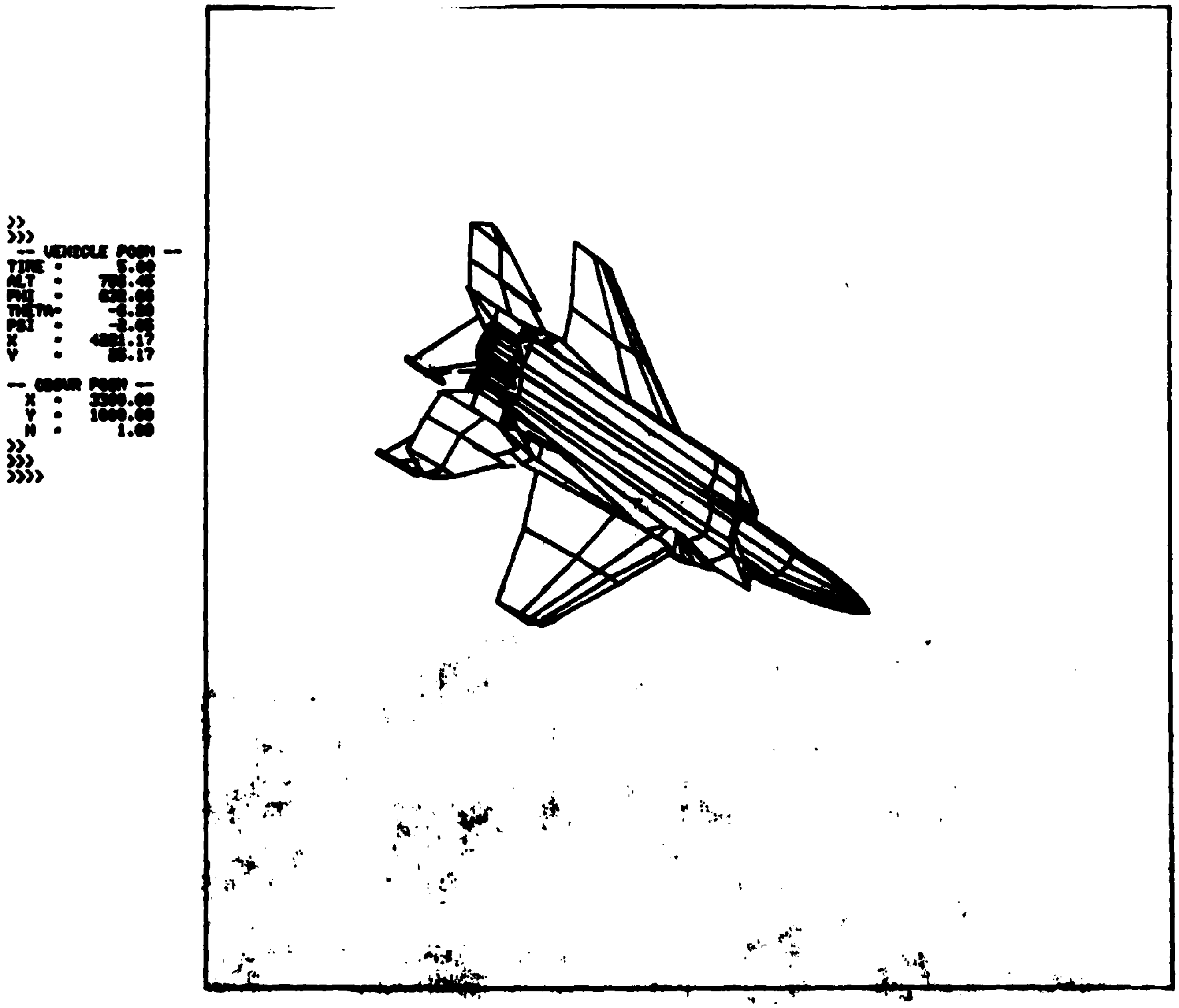


FIGURE 42
FIXED POSITION MODE - T=5

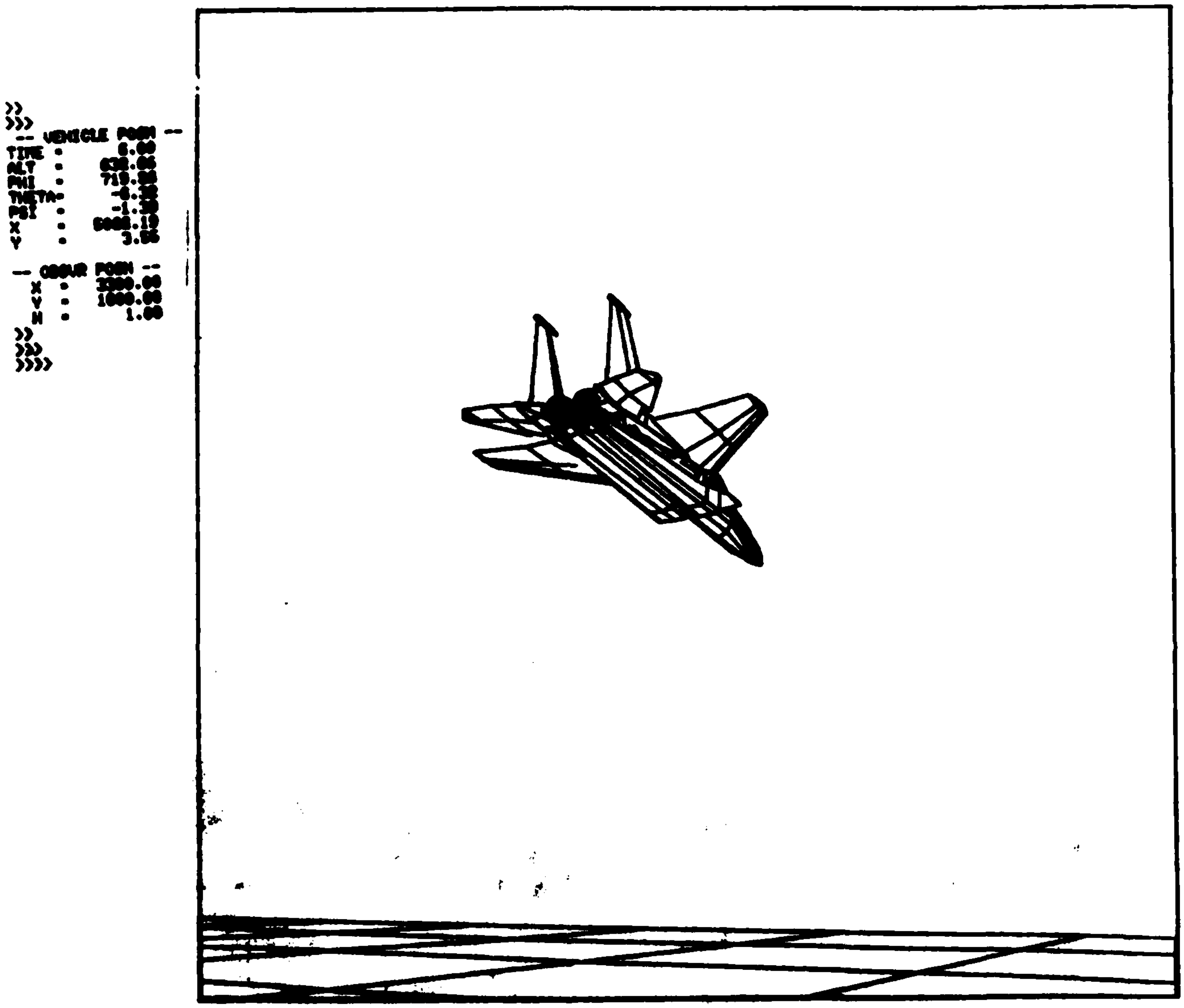


FIGURE 43
FIXED POSITION MODE - T=6

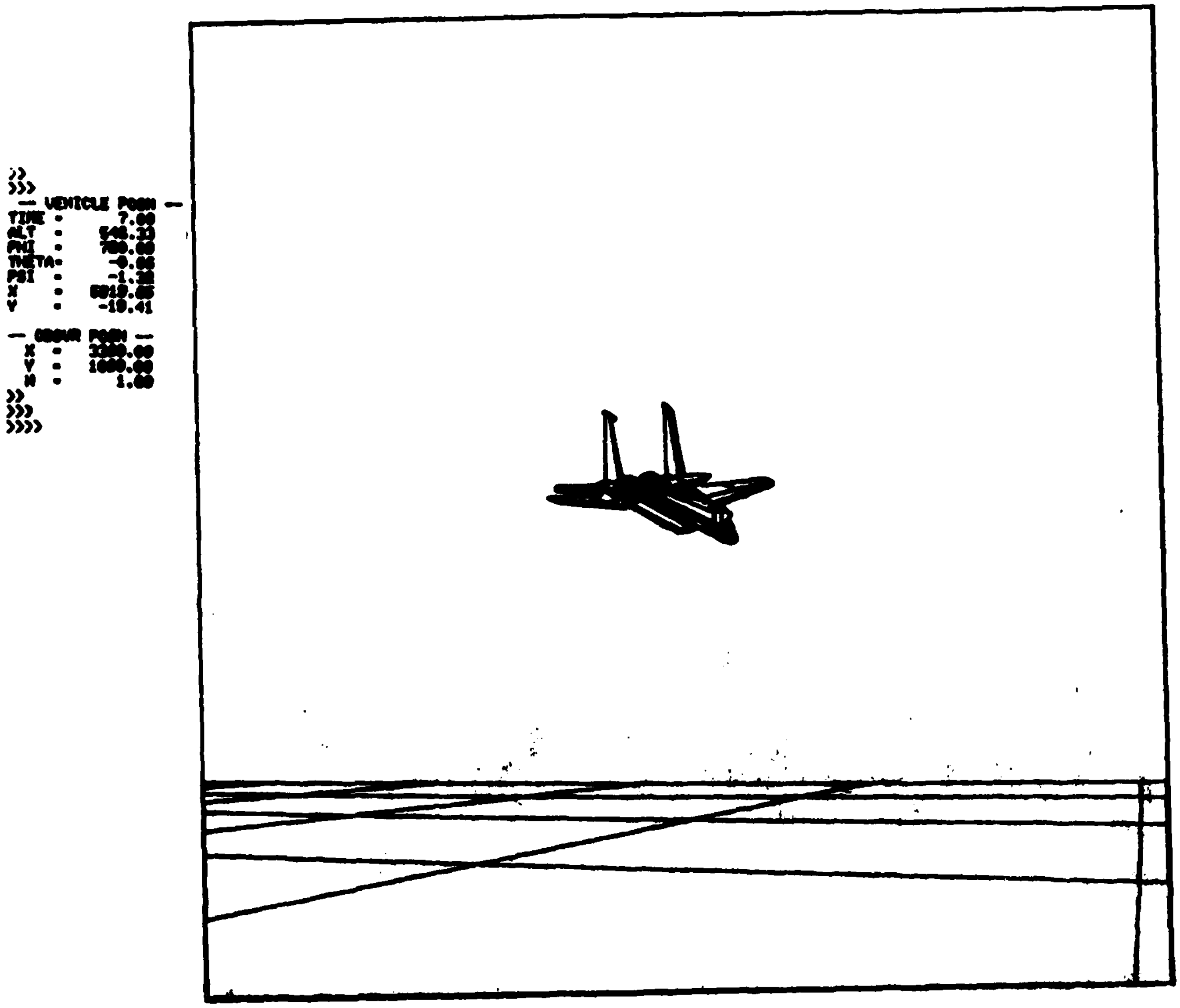


FIGURE 44
FIXED POSITION MODE - T=7

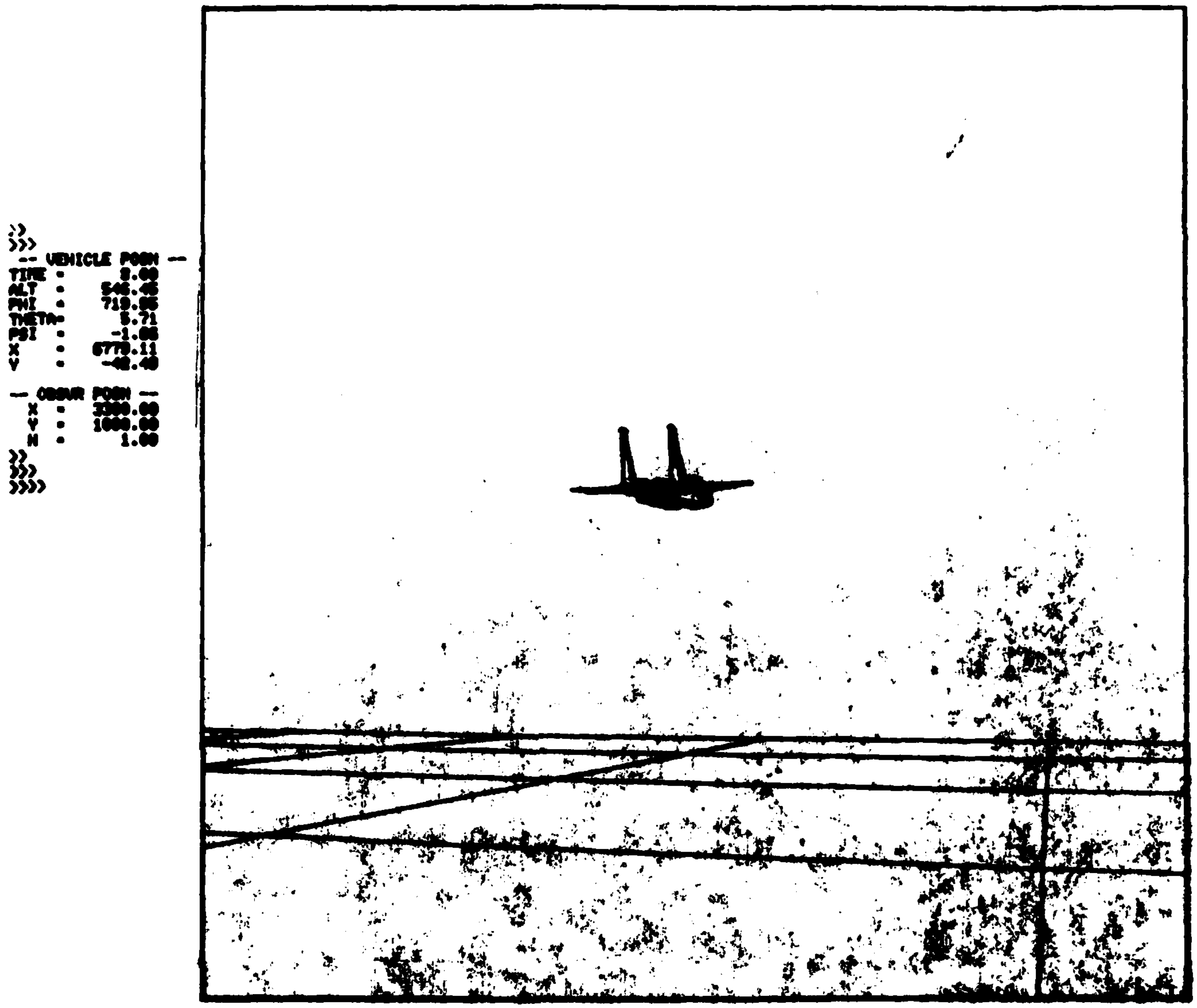


FIGURE 45
FIXED POSITION MODE - T=8

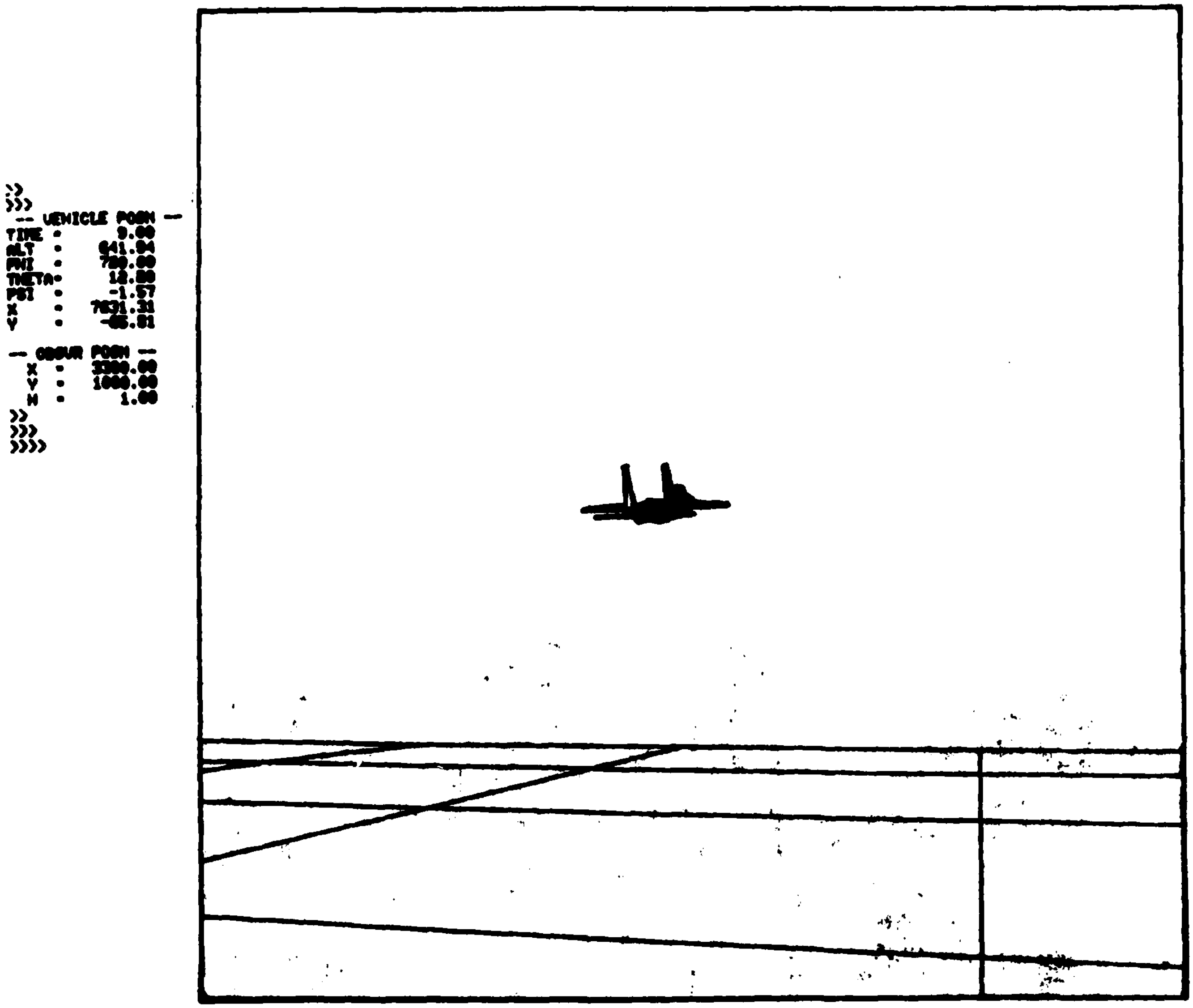


FIGURE 46
FIXED POSITION MODE - T=9

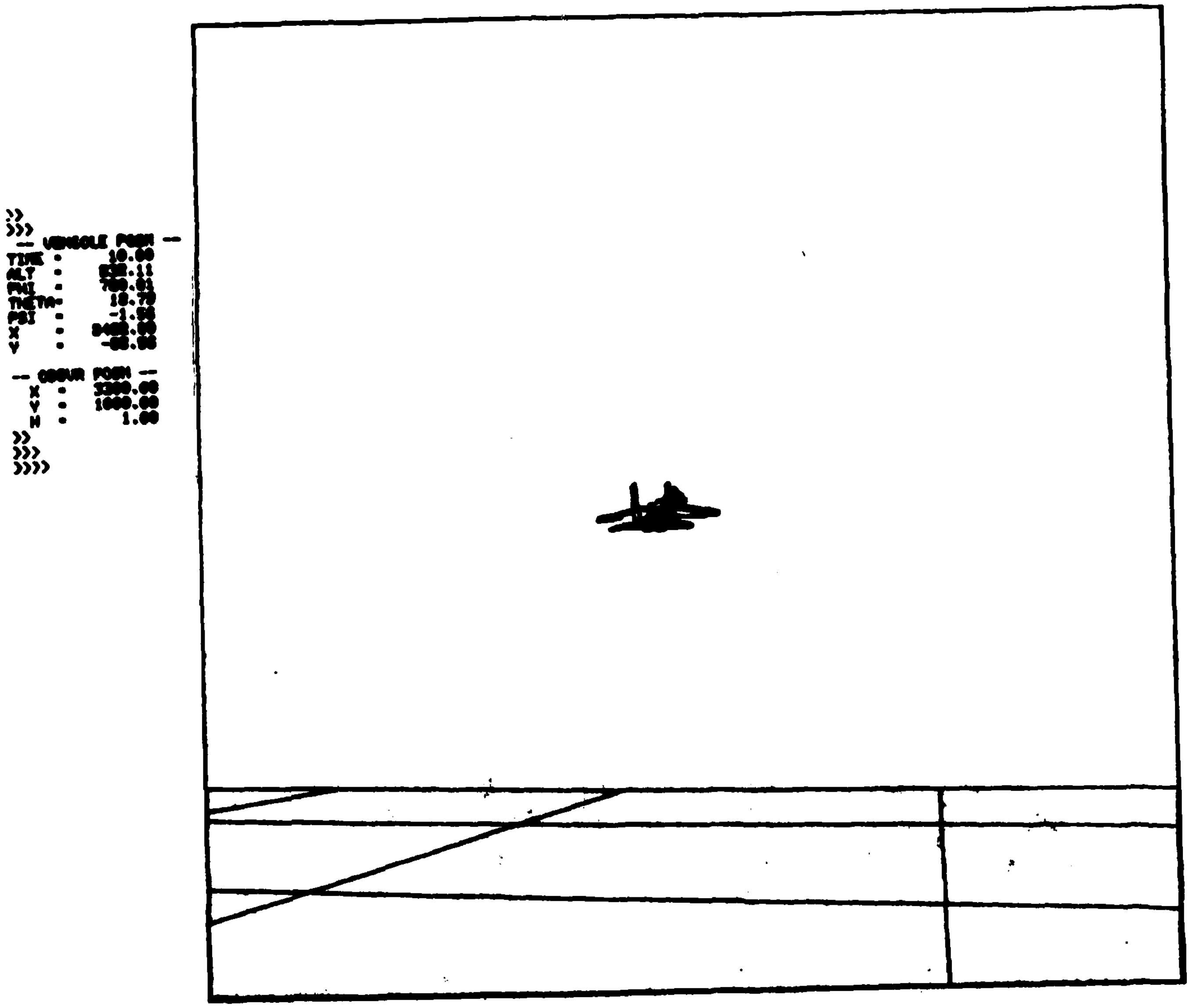


FIGURE 47
FIXED POSITION MODE - T=10

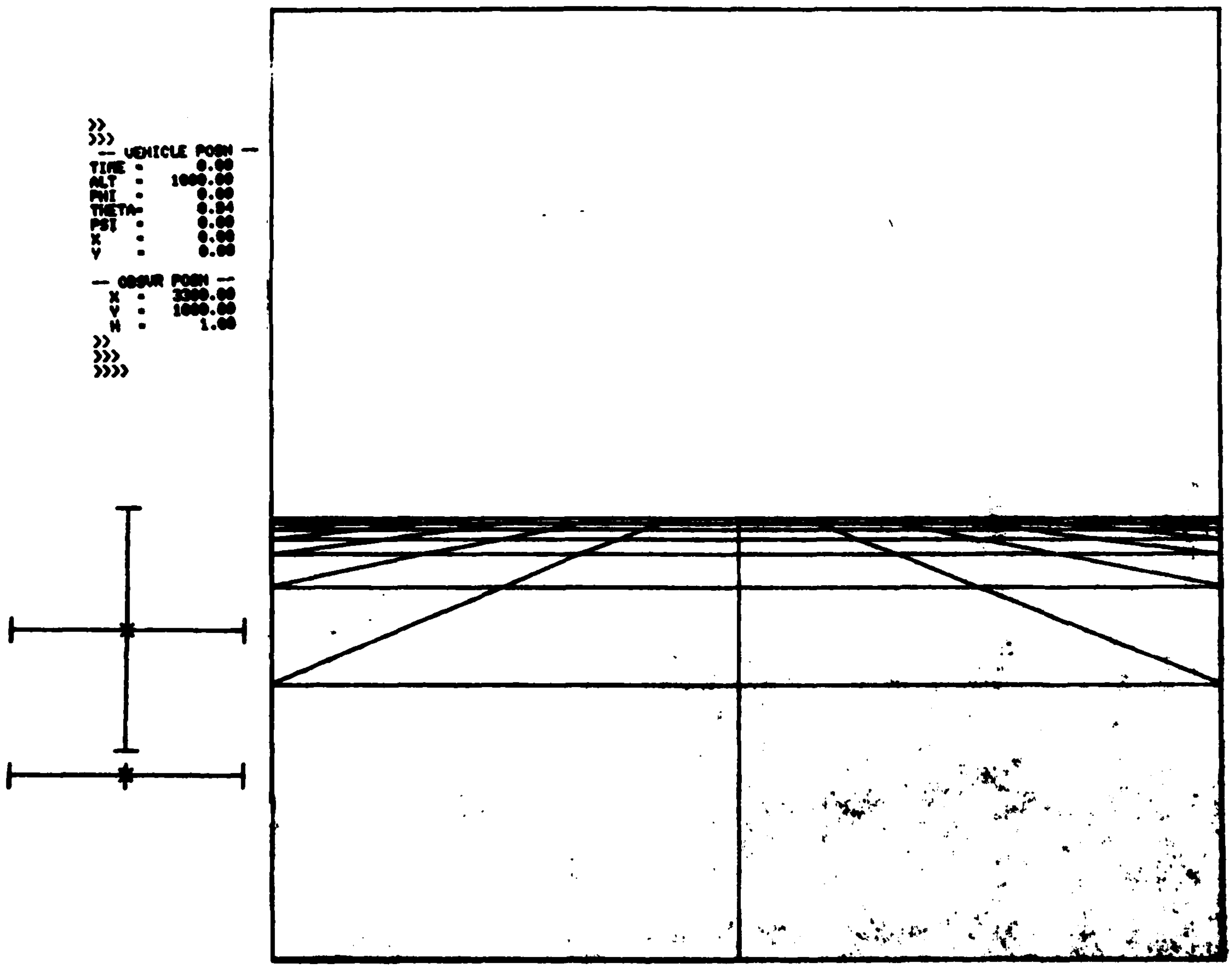


FIGURE 48
PILOT EYE MODE - T=0

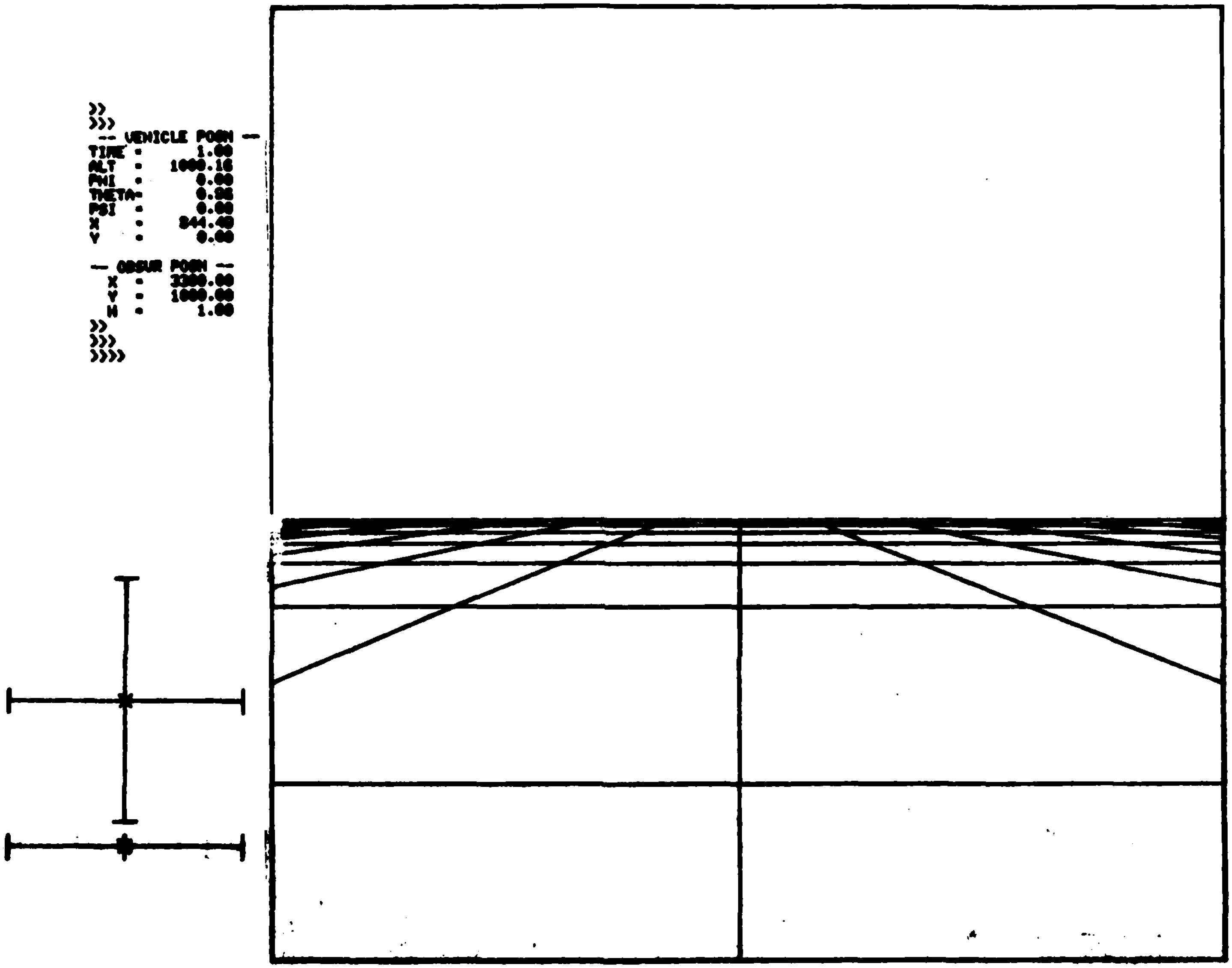


FIGURE 49
PILOT EYE MODE - T=1

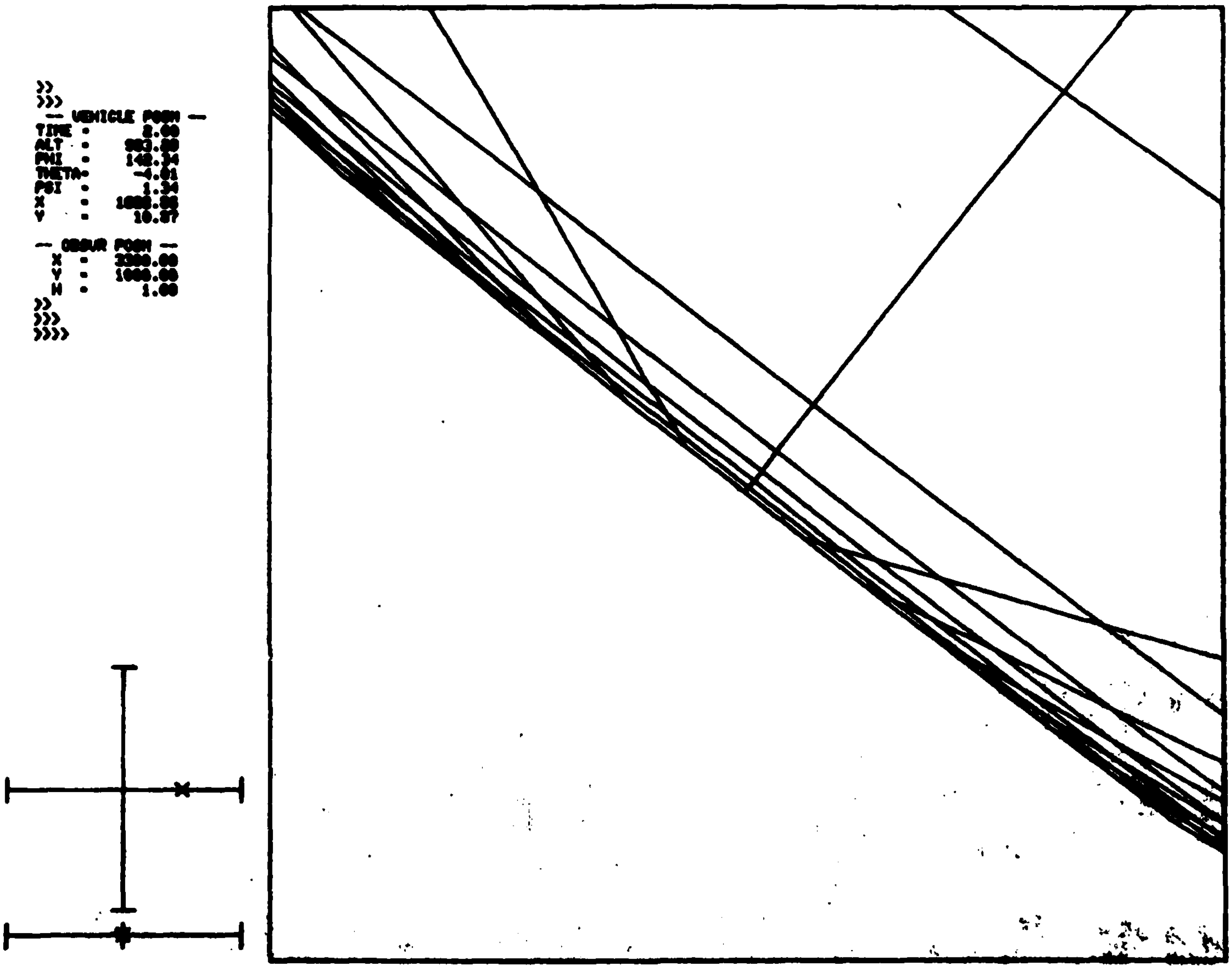


FIGURE 50
PILOT EYE MODE - T=2

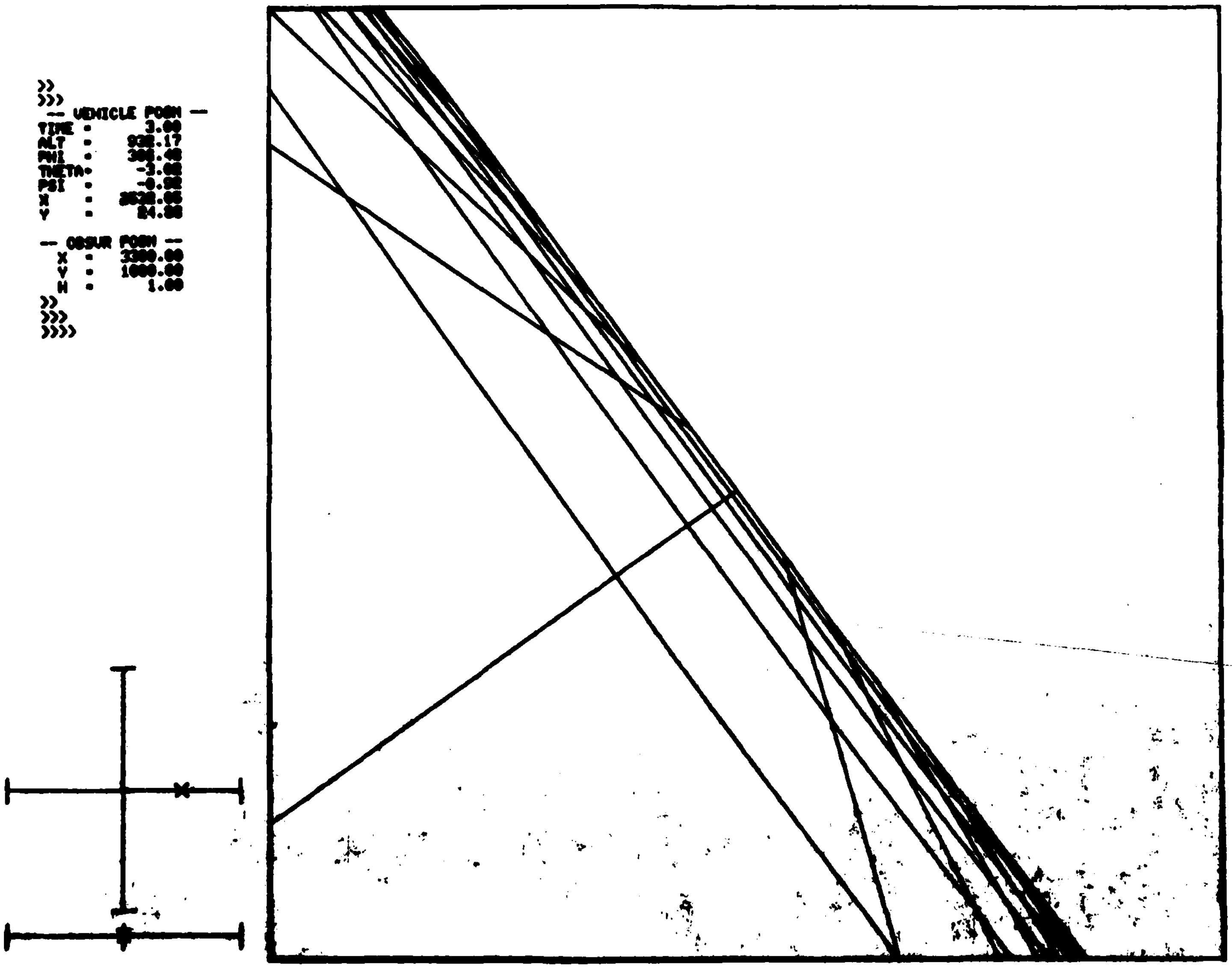


FIGURE 51
PILOT EYE MODE - T=3

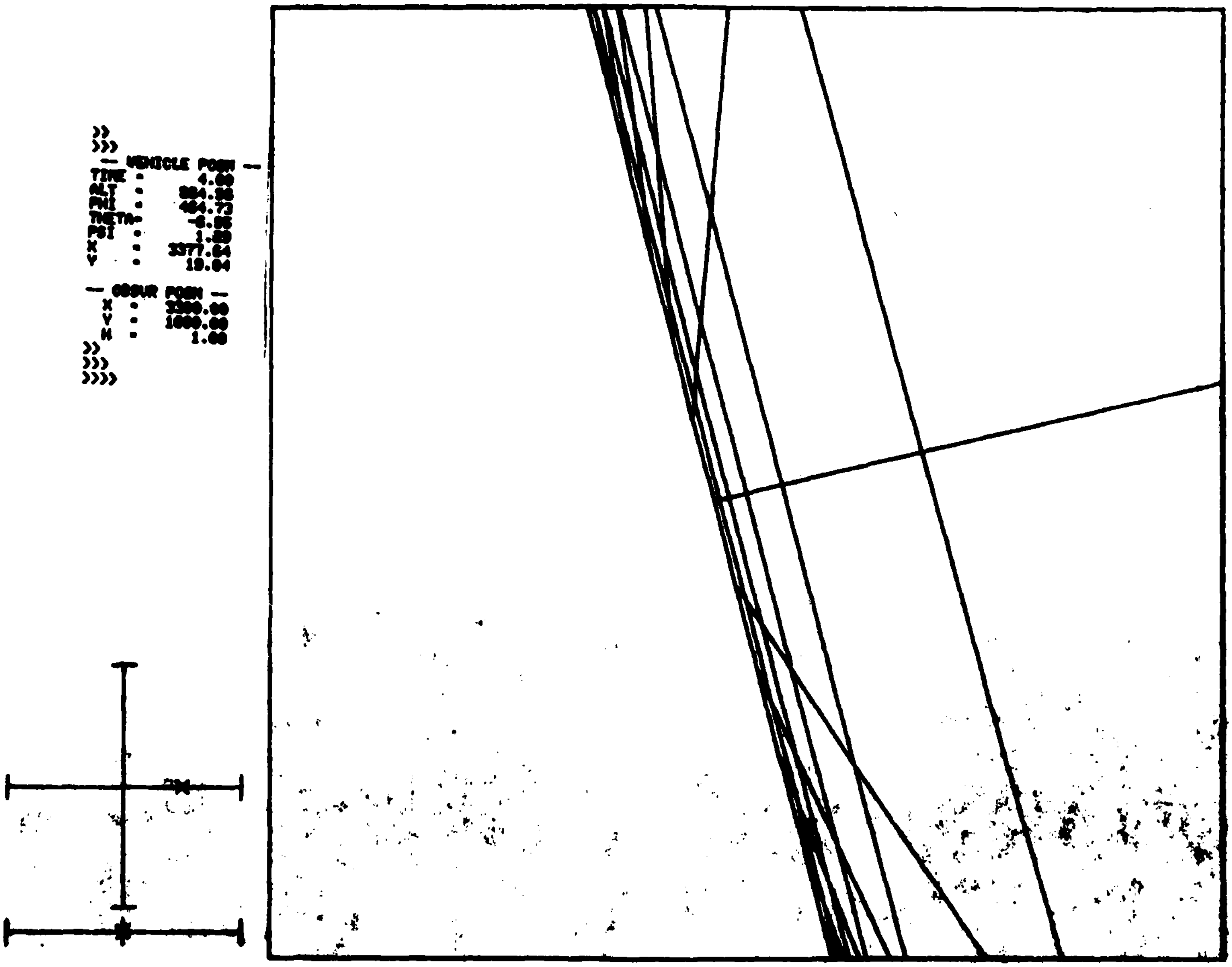


FIGURE 52
PILOT EYE MODE - T=4

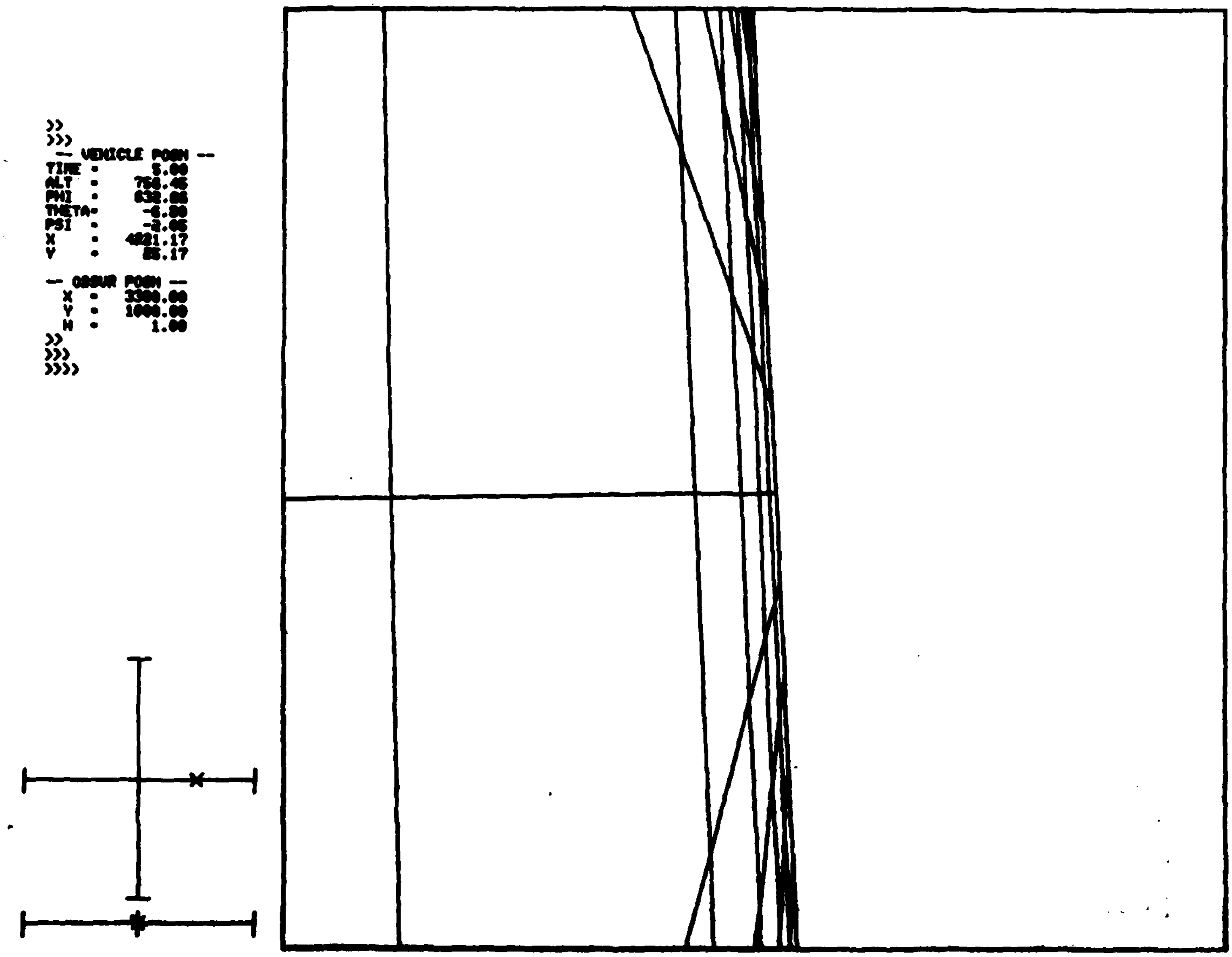


FIGURE 53
PILOT EYE MODE - T=5

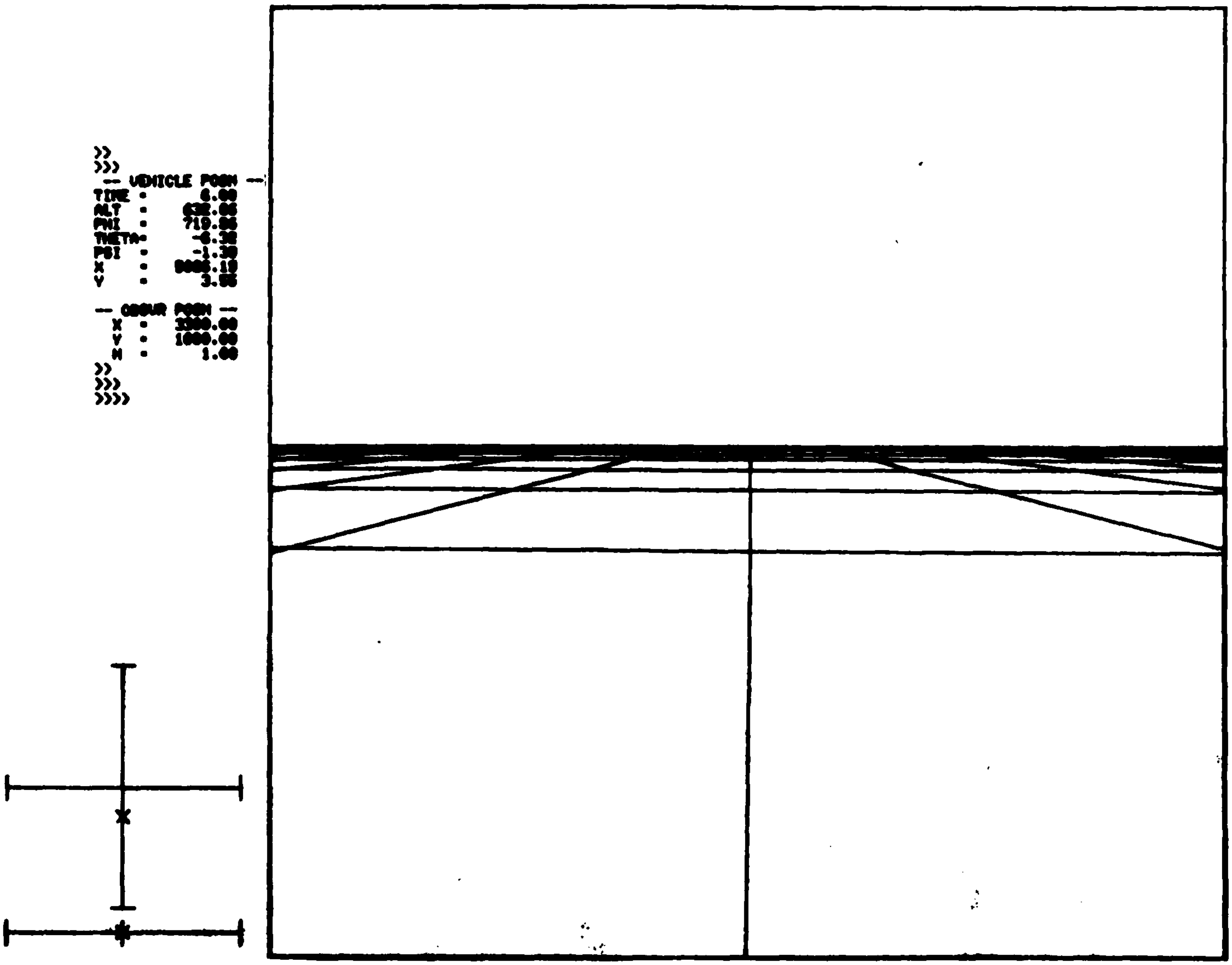


FIGURE 54
PILOT EYE MODE - T=6

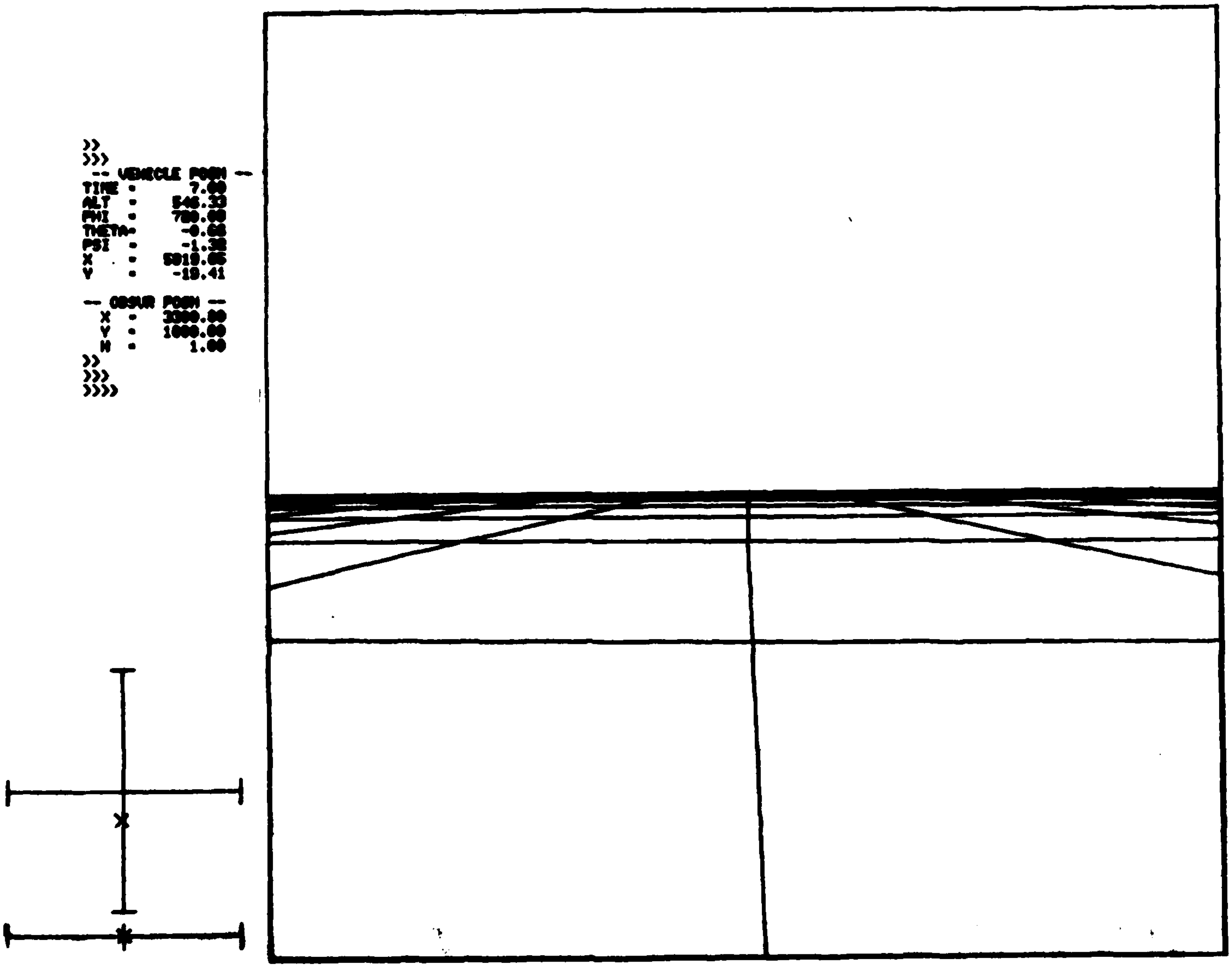


FIGURE 55
PILOT EYE MODE - T=7

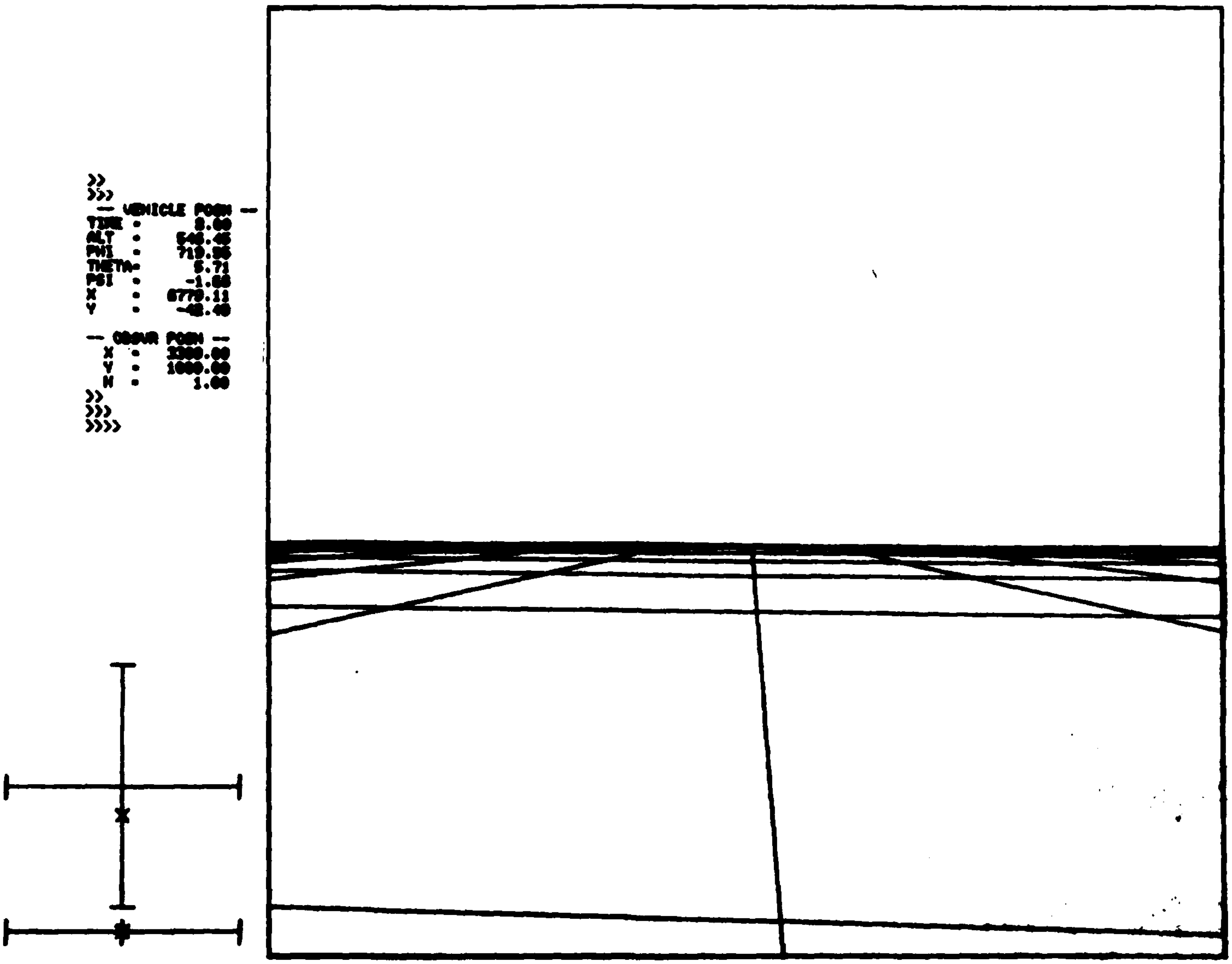


FIGURE 56
PILOT EYE MODE - T=8

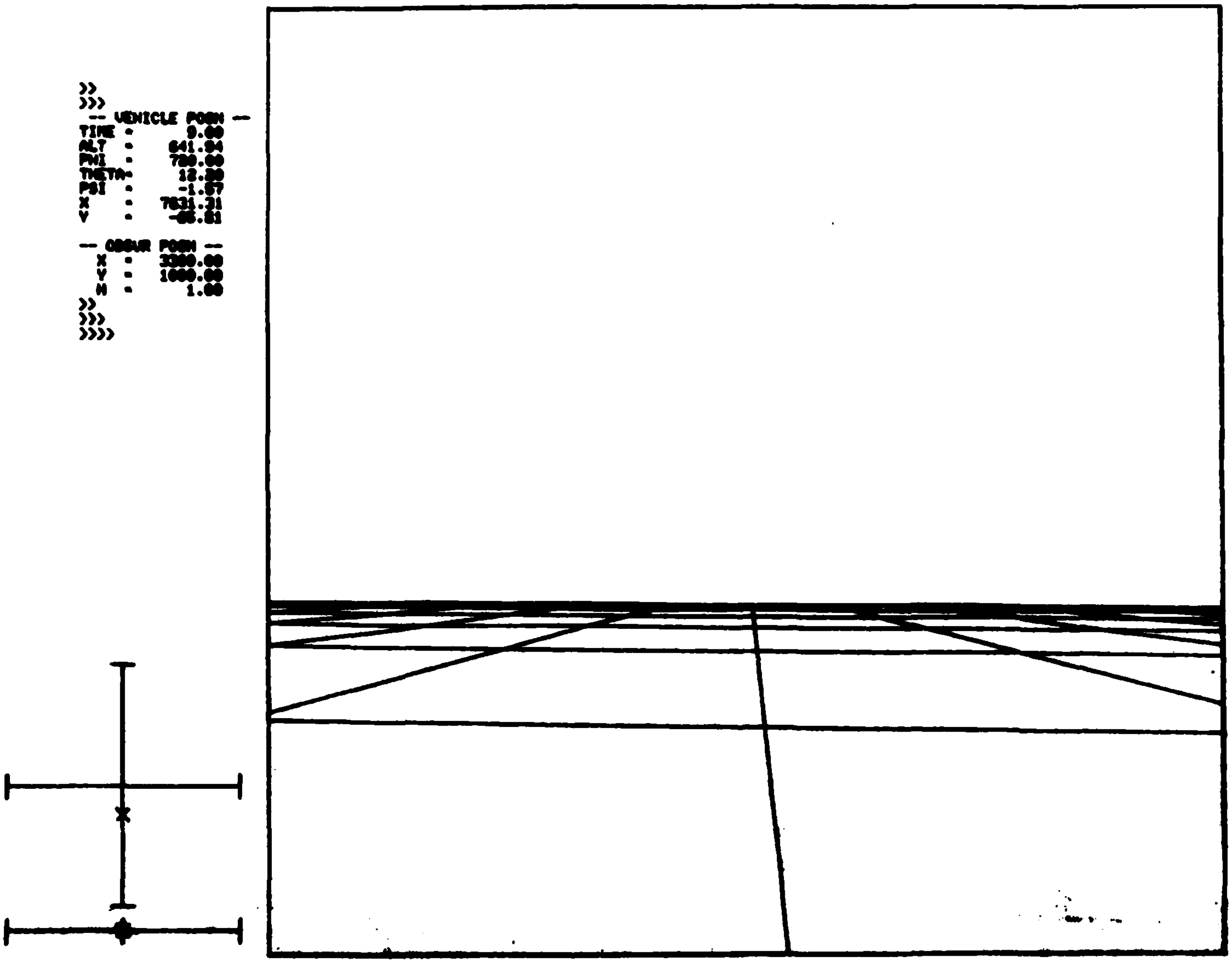


FIGURE 57
PILOT EYE MODE - T=9

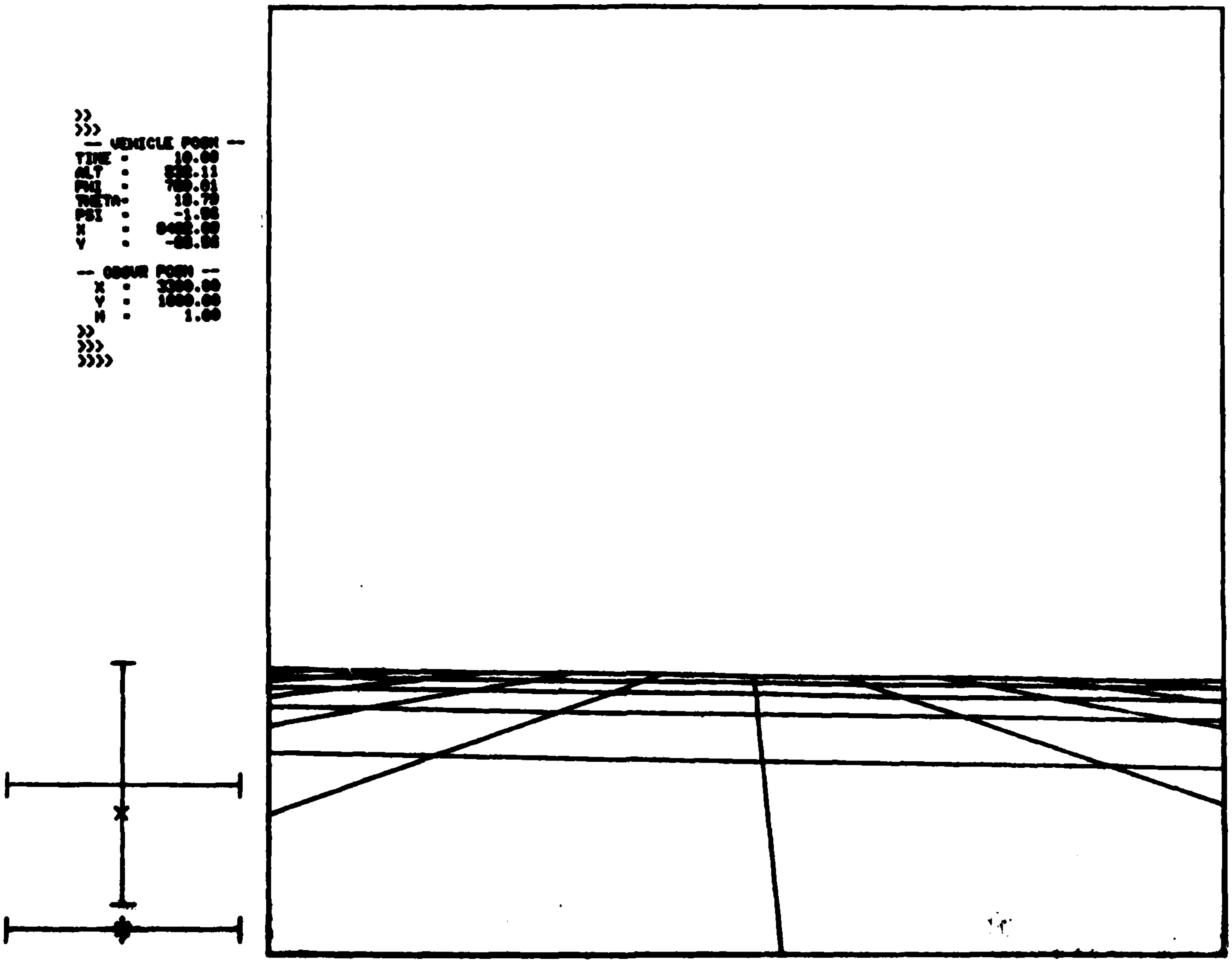


FIGURE 58
PILOT EYE MODE - T=10

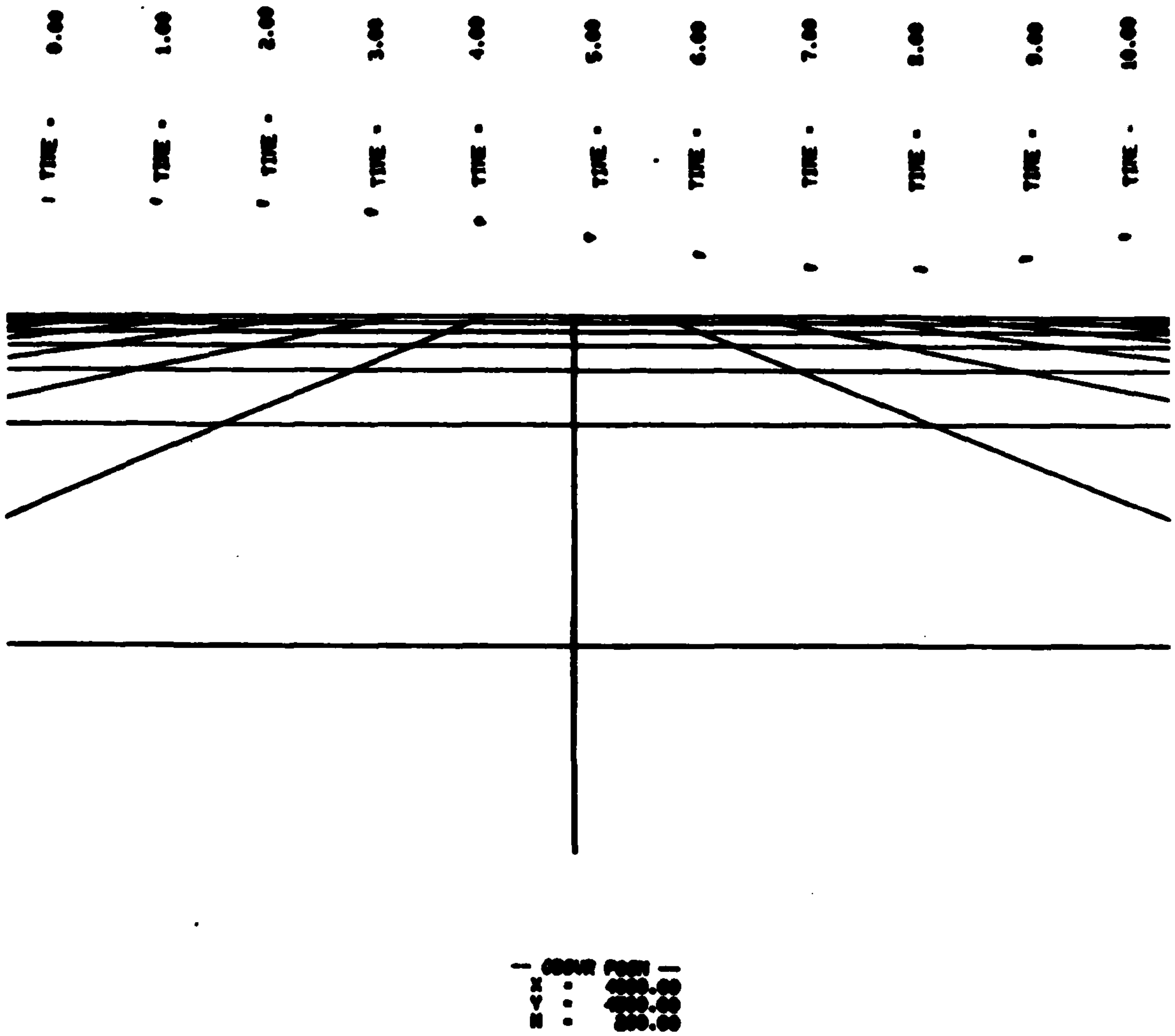


FIGURE 59
POSTER PLOT OF SAMPLE MOTION - SIDE VIEW

FIGURE 60
POSTER PLOT OF SAMPLE MOTION - FRONT VIEW

APPENDIX B

LESSONS LEARNED ABOUT ANIMATED MOVIES

Typically, engineers present data to other people through X-Y plots of the data. These plots can contain a great deal of information about a dynamic system and can be generated easily. CARTOONE can present this same data in a more easily understandable way by showing how the data translates into motions of a solid body. Plotting data requires very few decisions about how the information should be presented, since the available options are limited to what type of graph paper to use and how to calibrate the axes. When creating an animated movie with CARTOONE, the manner in which the data is presented is just as important as the data itself. Showing motions improperly detracts from the efficiency of a movie, and may even render it useless. We learned several lessons while creating movies, and this appendix addresses these problems.

CARTOONE assumes that the origin of the geometry model coincides with the center of rotation of each independently moving body, and manipulates them accordingly. If this is not true, the bodies will be rotated about a different center of rotation, and the motions will look wrong. This very simple rule is often forgotten by CARTOONE users. An aircraft rotates about its center of gravity (cg), and the body is symmetric about the X-Z plane, so the lateral position of the cg rests within this plane. A small error in the longitudinal position of the cg will not affect the appearance of the motion significantly since the body is long, and the small error will only change the cg position a small percentage of the body length. The same error in the vertical position of the cg might affect the appearance of the vehicle rotations quite a bit, especially in roll (i.e. rotation about the X axis), since an aircraft is also slender. The same error may result in a large percentage change along the vertical axis. Another related problem is that CARTOONE centers each picture on the center of rotation of the primary vehicle, thus making the body appear to rotate about the center of the screen. If the center of rotation is offset from its correct position, the body will be offset somewhat from the center of the picture, and will move around the screen as the body moves. This is undesirable from an aesthetic viewpoint. The user should always ensure that the vehicles are properly positioned with the framework of the geometry model.

Naturally, the aesthetics of a movie will affect how it is received by an audience. A movie that is aesthetically pleasing will be much more effective than one that is not since people are apt to pay closer attention to a movie that they enjoy. A boring movie will pass very little useful information to its distracted audience. The user of CARTOONE usually directs his film at a particular audience, so it is safe to assume that the audience has at least a passing interest in the subject of the film. However, even the most interested viewer can lose interest in a movie if it bores him. Users tend to use a series of Title Pages to present a large amount of quantitative data or a description of the motion to be shown. Although some data and descriptions may be necessary, too much becomes tedious. The audience may let their attention stray away from the screen, thus missing some of the motion. The purpose of a motion is to show motion of a dynamic system, not give a detailed description of the system. The number of Title Pages should be minimized. A good rule of thumb is to average three Title Pages per film clip. This should be sufficient to present enough information to define what is to be shown. Each Title Page is shown for six seconds, and a typical motion segment will take 20 seconds. If four Title Pages precede the motion, the Title Pages will be on the screen longer than the motion. Animated movies should primarily provide a visual description of a motion. The quantitative data should be of secondary importance.

Another pitfall of Title Pages is the temptation to end the movie with a series of Title Pages containing credits or conclusions. The audience usually will let its attention wonder from the screen once the final motion finishes, and the information being presented in the final few Title Pages will be lost. If the motions are sequenced properly, the conclusions to be drawn from the movie should be obvious. To maintain the interest of the audience, keep all credits at the beginning of the movie. It is usually desirable to begin the movie with an animation also, rather than opening with a Title Page. This attracts the attention of the audience right away, and gives the audience a chance to focus on the screen. Once again, the motions are the purpose for creating a movie with CARTOONE, and Title Pages should be used to enhance the motions, not detract from them.

The length of a movie will vary with the amount of material being presented. A CARTOONE user will probably be presenting technical data, and the typical viewer cannot absorb an infinite amount of data at one time. We have found that a movie dealing with a complex technical subject should not exceed 10-12 minutes. Any longer and some of the audience will not be able to ingest all the information being presented. Furthermore, if the entire film is covering a single subject, a longer film can become tedious.

We have had more use for the Fixed Position mode than any other view mode, since it is very useful in showing a motion as seen by a typical observer. However, it has several drawbacks. As an aircraft moves past an observer, the error between the observed attitude and the actual aircraft orientation changes. This error, known as parallax, changes with observer position as well as aircraft position. Two observers do not see the same thing, even though they are watching the same series of events, since they are at different positions in space and have different parallax errors. To re-create the action seen by a particular observer, the observer location must be known fairly exactly. For this reason, this view mode is unsuitable for illustrating aerodynamic characteristics, and especially unsuitable for illustrating incremental changes due to configuration changes. The parallax changes so much for this mode that the vehicle motions are obscured by parallax effects. This is also true to a lesser degree for the Chase Plane Mode. The parallax changes are less for this mode than with a Fixed Position View, but it is still present. The Wingman Mode performs this task very well, and should be used to illustrate aerodynamic characteristics and incremental effects.

Many rapidly developing motions cannot be fully comprehended at real-time speeds, and must be shown in slow motion. However, a motion should never be shown only in slow motion because the audience will not get a proper feel for how rapidly the motion actually does develop. This decreases the efficiency of the movie, especially for motions in which some sort of human interaction is required. If the audience does not see the true speed of the motion, they may get an erroneous impression of the magnitude of the task. We face this problem quite often when dealing with aircraft pilots. To properly demonstrate such a motion, it should first be shown with an external view mode of real-time speed, then with an external view mode in slow motion. If applicable, a real-time Pilot Eye View should then be shown, thus emphasizing how fast the motion develops.

When an aircraft flies past an observer, it changes its apparent orientation. In real life, an observer turns his head to follow the aircraft. As it approaches, it is pointed toward the observer, and as it leaves, it is pointed away from the observer. It changes its observed direction of flight as seen by the observer, even though its absolute direction of flight is constant. This effect appears in CARTOONE also. Figure 61 shows a Poster Plot of an F-4 Phantom flying beneath an observer just before it passes, just as it passes, and just after if passes. Figure 62 shows the same motion from the Fixed Position Mode. As it passes, it appears to change its direction of flight. This apparent change in direction comes from the centering statements, just as in real life the apparent change comes from the observer turning his head. Although this change is identical to a real world effect, it is not desirable to see an aircraft abruptly change direction in a movie, as happens if it passes directly over or beneath the observer. Keep the observer away from the plane of motion, so that the change in direction appears gradually, not instantaneously.

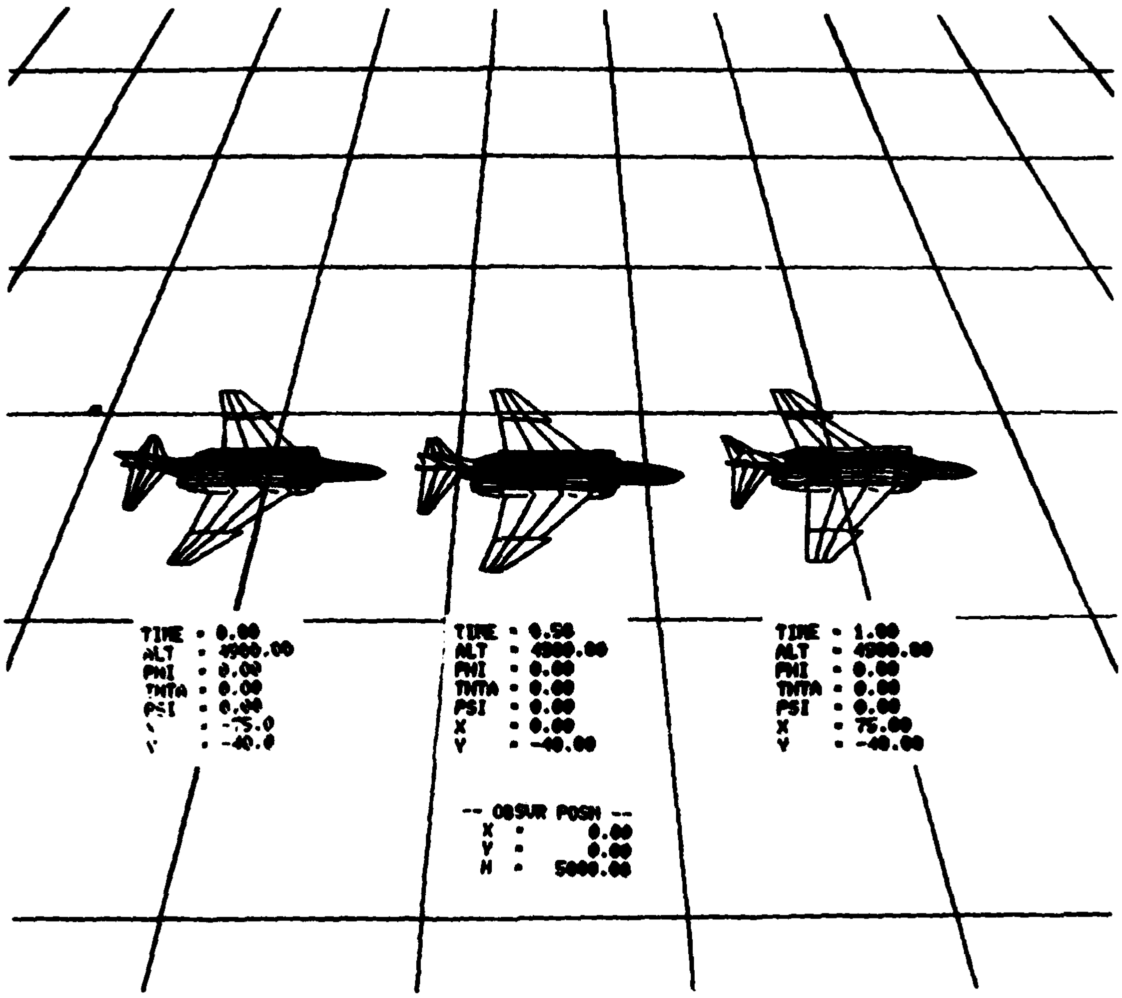


FIGURE 61

AIRCRAFT FLYING BENEATH OBSERVER - POSTER PLOT

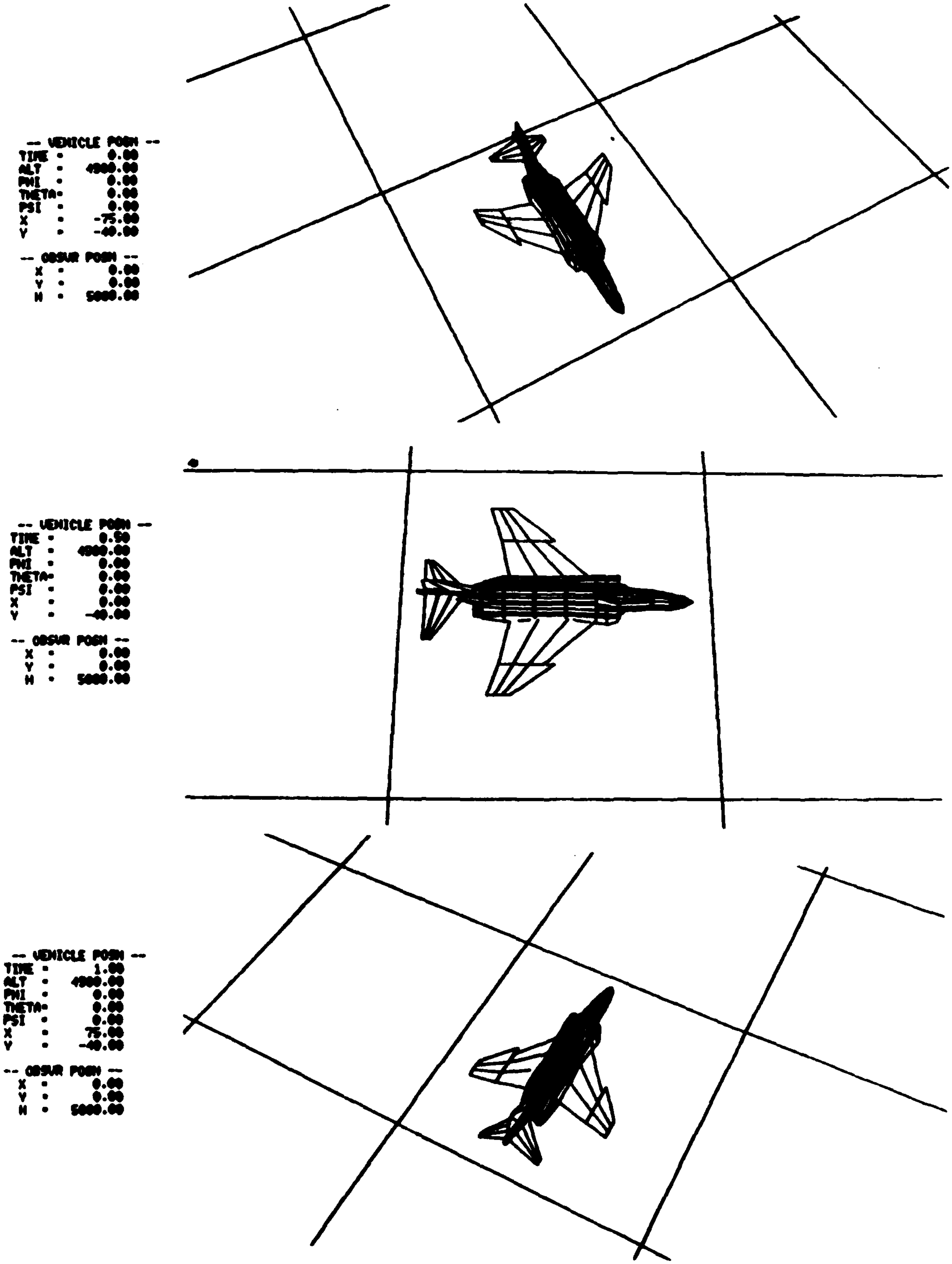


FIGURE 62

AIRCRAFT FLYING BENEATH OBSERVER - FIXED POSITION MODE

REFERENCES

1. Henry Christiansen, et.al., "MOVIE.BRIGHAM YOUNG UNIVERSITY A General Purpose Computer Graphics System," January 1980.

2. Jan Roskam, "Airplane Flight Dynamics and Automatic Flight Controls," Roskam Aviation and Engineering Corporation, 1979.

3. Publication Number 90598, "Universal Pagesetter Format - Operator's Manual," Information International, Inc., February, 1980.

* U.S. GOVERNMENT PRINTING OFFICE 1980 - 659-060/85